Apuntes del Curso



Organizado por: ***Comisión Nacional de Energía Atómica***
Distribuidor Nacional CDS/ISIS

Fecha: ***19 al 23 de Julio de 1999***

Capacitadores: ***Marcia Ymanaka Barretto*** **- Bireme**
Vinicius de Andrade Antonio **- Bireme**

Composición y redación: ***Deharbe Hipólito*** *(deharbe@infovia.com.ar)*
Lopez Marcos *(mlopez@fceco.uner.edu.ar)*
Revisión: ***BIREME***

El presente trabajo intenta reflejar los contenidos abordados en el Curso de WXIS (WWWIsis XML IsisScript Server). Para la comprensión de los temas, se han abordado una serie de ejercicio en los que se detallan los pasos seguidos.

El paradigma que se plantea difiere de las aplicaciones hasta ahora realizadas con el WWWIsis. La idea es abordar a través de una Script, todas las acciones que se desean realizar. Por lo tanto, el esquema de trabajo de aquí en adelante, se abocará a la construcción de la Script para interactuar con los formularios en el Web.

Para este trabajo se utiliza el software *wxis.exe* en entorno cgi-bin. La versión:

```
CISIS Interface v3.45_3/PC32/M/30000/10/30/I - XML IsisScript 1.0
CISIS Interface v3.45_3/.iy0/Z/4GB/GIZ/DEC/INVX/B64/FAT/CIP
Copyright (c) BIREME/PAHO 1999. All rights reserved.

WXIS release date: 19990716
```

WXIS:

- No es un programa residente
- Es activado cada vez que se lo llama.
- No almacena por sí mismo un estado.
- Es dirigido o programado por las Scripts del WXIS.
- Antes se poseían varios archivos cgi, ahora es sólo necesario uno: la Script.
- Está disponible para varias plataformas (Windows 95, NT, Unix, etc.)
- Trabaja con un archivo de licencia. La versión que se utiliza en el curso puede ser usado en ambiente de prueba.

Se monta una Script por lo que el WXIS va a realizar.

Desde del web, el cliente invoca la ejecución de una script WXIS a través del cgi para actuar sobre la base de datos ISIS, el resultado obtenido dependerá del propósito de la script pudiendo ser una página con los resultados de una búsqueda, la creación o modificación de un registro, actualización del archivo invertido, etc.

También puede ser ejecutada desde la línea de comandos del Sistema Operativo.

Toda IsisScript tiene una estructura que se debe respetar. La misma es la siguiente:

```
<IsisScript name=nombre_del_Script>
        <section name=nombre>

                ....... Área de tareas ........
```

```
        </section>
    </IsisScript>
```

A continuación se llama al WXIS a través del prompt del sistema. Antes de invocarlo se confeccionó una Script, que se denominó **X.XIS**:

```
1. <IsisScript name=Hello>
2.     <section name=um>
3.         <display>Hello World</display>
4.     </section>
5. </IsisScript>
```

Se puede apreciar la estructura planteada con anterioridad. En la línea 3, para poder desplegar el mensaje *Hello World* se debe colocar el mismo entre las marcas *<display>* y *</display>*.

Se lo invoca desde el prompt del sistema de la siguiente manera:

C:\Curso>wxis IsisScript=x.xis

Es importante respetar las mayúsculas y minúsculas de la palabra IsisScript.

Luego de la ejecución se visualiza el mensaje *Hello World* en la pantalla.

Si se desea utilizar un formato Isis para la visualización, el mismo deberá ser indicado entre las marcas *<pft>* y *</pft>*. El caso anterior queda planteado:

```
<IsisScript name=Hello>
    <section name=um>
        <display>
        <pft>'Hello World'/,'Esta es una segunda línea'</pft>
        </display>
    </section>
</IsisScript>
```

También se puede recurrir a un formato externo, en ese caso, dentro de las marcas <pft> se deberá colocar el nombre del formato precedido por el símbolo @.

Ej. <pft>@formato1.pft</pft>

donde @formato.pft puede contener

'Hello World'/,'Esta es una segunda línea'

De aquí en más, para todos los ejercicios, se trabaja con el Servidor Web OmniHTTPD. El directorio por default es C:\httpd\HtDocs. La base de datos es CDS ubicada en el directorio C:\Curso. Los archivos correspondientes a las Script

se alojan en el directorio C:\httpd\cgi-bin\ar\. Los formularios se guardan en el directorio C:\httpd\HtDocs\ar.

Para verificar si funciona el WXIS.EXE, se puede enviar una llamada al mismo por medio del browser, utilizando la siguiente sintaxis:

http://localhost/cgi-bin/wxis.exe/?hello

El sistema responde con la siguiente pantalla:



A continuación se plantea el primer ejercicio que consiste en desplegar en el Browser, el mensaje HELLO WORLD!.

Ejercicio Nº 1

A través un formulario se invoca al WXIS para que despliegue en una página Web el mensaje **Hello World!.**

El formulario en html se lo ha denominado **Ejerc1.html** y tiene la siguiente codificación:

```
1.  <html>
2.  <head>
3.     <title>Script 0</title>
4.  </head>
5.  <body>
6.        <form action="/cgi-bin/wxis.exe" method="post">
7.            <input type="hidden" name="IsisScript" value="ar/script0.xis">
8.            <input type=submit value="OK para ver Hello...">
9.        </form>
10. </body>
11. </html>
```

En la línea 6 se le indica al formulario que debe invocar al WXIS.EXE que se encuentra en el directorio cgi-bin. El método utilizado es post. Este línea será un constante en todas las llamadas al WXIS.

La línea 7 especifica a través de un campo oculto, que el WXIS va a utilizar una IsisScript cuyo nombres es **script0.xis** y se encuentra en el directorio **ar** (dentro de HtDocs).

La línea 8 corresponde al submit (botón) que permite enviar los datos del formulario.

Cuando se carga en el Browser el archivo **ejerc1.html**, se puede observar la siguiente pantalla:



El archivo **script0.xis** tiene la siguiente codificación:

```
1.  <IsisScript>
2.        <section>
3.              <display><pft>'Content-type: text/html',/#</pft></display>
4.              <display>Hello World</display>
5.        </section>
6.  </IsisScript>
```

En la línea 3 se indica al WXIS que el resultado lo debe devolver en formato html para que sea visualizado por el browser. Ello se indica dentro de marcas *<pft>*.

Luego de hacer click sobre el botón “OK para ver Hello...”, se observa la siguiente pantalla:



Dentro del mismo ejercicio, se intenta pasar desde el formulario el nombre del cliente. Para ello, al formulario se le debe incorporar una nueva línea para tomar los datos del usuario. La codificación del mismo es:

```
1.  <html>
2.  <head>
3.      <title>Script 0</title>
4.  </head>
5.  <body>
6.      <form action="/cgi-bin/wxis.exe" method="post">
7.              <input type="hidden" name="IsisScript" value="ar/script0.xis">
8.              Ingrese su nombre <input type="text" name=nombre><br>
9.              <input type=submit value="OK...">
10.     </form>
11. </body>
12. </html>
```

En la línea 8 se observan las marcas html para ingresar el nombre del usuario que luego será pasado a la Script a través de la variable *nombre*.

La visualización en el browser:



El archivo **script0.xis** queda codificado:

```
1.  <IsisScript>
2.      <section>
3.          <!-- a continuación se almacena en el tag 13 el contenido de la
            variable nombre que fue pasado a través del cgi -->

4.          <field action="cgi" tag="13">nombre</field>
5.          <display><pft>'Content-type: text/html',/#</pft></display>
6.          <display><p><center><u><h1>Hello
            World!!!</h1></u></center></p></display>

7.          <!-- muestra el contenido del tag 13, que fue almacenado
            anteriormente -->

8.          <display><pft>'Su nombre es: ',v13</pft></display>
9.      </section>
10. </IsisScript>
```

En la línea 3 y 7 a través de las marcas *<!--* y --> se pueden colocar comentarios.

En la línea 4, se le asigna a una variable de entorno cgi (campo v13) el dato que fue cargado en el formulario y almacenado en la variable *nombre*.

La línea 8 muestra la leyenda “Su nombre es:” y visualiza el contenido del tag 13. Tener en cuenta que cuando se desea visualizar el contenido de alguna etiqueta, dicha información debe estar encerrada por las marcas *<pft>* y *</pft>*.

El browser devuelve la siguiente pantalla:



Tareas en WXIS.

Las tareas en WXIS se indican dentro de las marcas *<do>* y *</do>*. La estructura de la Script es:

```
<do task=...>

    <parm name=…> … </parm>
    <parm name=…> … </parm>

    <loop>

    </loop>

</do>
```

Los parámetros indicados dependerán de la tarea (task=) a realizar.

Por ejemplo, con la siguiente Script(**x1.xis**) se visualizan los siguientes 10 registros correspondientes al MFN número. 25. La misma está preparada para ejecutarse desde el prompt del sistema, en el directorio donde se encuentra la base.

```
1.  <IsisScript>
2.      <section>
3.      <display>Base de datos CDS</display>
4.          <do task=mfnrange>
5.          <parm name=db>CDS</parm>
6.          <parm name=from>25</parm>
7.          <parm name=count>10</parm>
8.              <loop>
9.                  <display><pft>#,mfn(5),</pft></display>
10.             </loop>
11.         </do>
12.     </section>
13. </IsisScript>
```

La línea 4 indica que la tarea(task) que se va a realizar es un *mfnrange*; es decir, una tarea sobre un rango de mfn's.

En la línea 5, a través del parámetro *<parm name=db>* se le indica a la Script el nombre de la Base de datos, para este caso: CDS.

El parámetro especificado en la línea 6, indica a partir de qué número de registro debe ser considerado el rango a procesar.

El parámetro de la línea 7 indica la cantidad de registros que deben ser procesados.

Notar que estos parámetros están definiendo la característica de la tarea 'mfnrange'.

Como se deben procesar varios registros, lo que se deba hacer por cada uno de ellos, debe indicarse dentro de un ciclo que se corresponde con las marcas *<loop>* y *</loop>*. Esto se refleja en las líneas 8, 9 y 10. El formato indicado entre las marcas *<pft>* y *</pft>* visualiza únicamente el MFN.

La línea 11 cierra la tarea con la marca *</do>*.

Nota: las marcas en la Script van de a pares: *una de apertura y otra de cierre.*

<.....> y </.....>

La ejecución del Script en el prompt:



Resumiendo:

La estructura de la Script para el entorno Web, se codifica:

<IsisScript>
<section>
<display><pft>'Content-type: text/html'/#</pft></display>

<do task= ... >
<parm name= .. > </parm>
<parm name= ...> </parm>
<parm name= ...> </parm>

Los parm name definen la característica de la tarea (task=)

<loop>

Dentro del loop, se accede a los datos del registro; por cada ocurrencia del ciclo se apunta a cada uno de los registros recuperados.

</loop>
</do>

```
    </section>
</IsisScript>
```

Como se mencionó, los *<parm name=...>* quedan sujetos a la tarea (task) definida. Para mayor información consultar el manual de referencia (reference.html).

Se puede tener el nombre de la base en un campo y con un formato referenciarla.

Ej. *<parm name=db><pft>*v32*</pft></parm>*

(1) (2)

Si entre (1) y (2) no hay texto, la Script ignora la línea.

Nota: entre las marcas de apertura y de cierre, siempre debe haber "algo".

A continuación se planeta la Script (**scrip1-1.xis**) que muestra en el Web, todos los registros de la base CDS visualizando el MFN, las etiquetas 24 y 70.

```
1.  <IsisScript>
2.      <section>
3.        <display><pft>'Content-type: text/html'/#</pft></display>
4.          <do task=mfnrange>
5.              <parm name=db>c:\curso\cds</parm>
6.                <loop>
7.                    <display><pft>'<b>',mfn(5)/,'</b><br>',v24/,'<br>',
                      (v70/),'<br><p>'/#</pft></display>
8.                </loop>
9.            </do>
10.     </section>
11. </IsisScript>
```

En la línea 5 se puede observar que para este caso se ha especificado la ruta o path en donde se halla la base de datos.

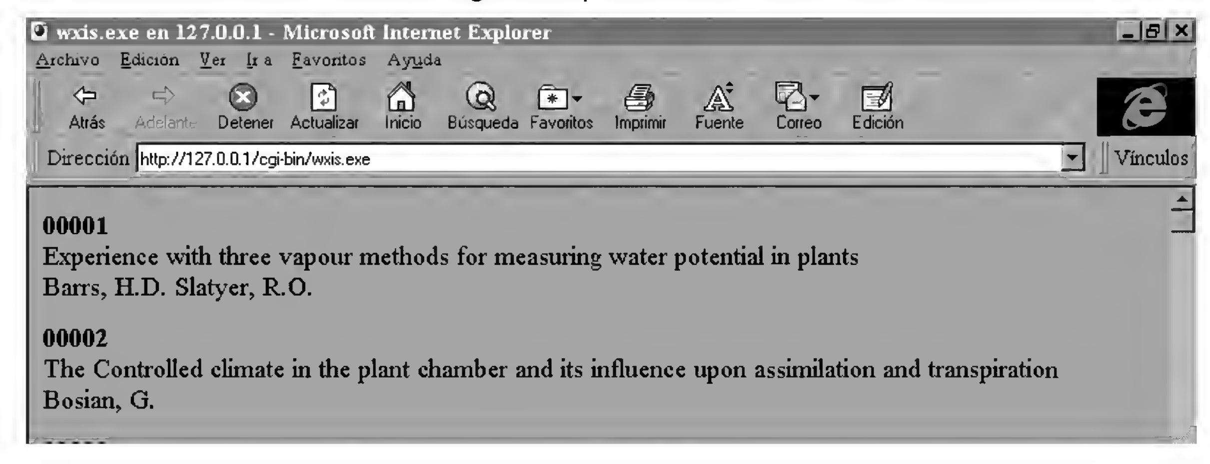
La línea 7 dentro del formato contiene literales que corresponden a marcas html dado que el resultado o display se realiza en el browser. Esta instrucción se encuentra dentro del loop; esto significa, que se va a aplicar a cada registro que procese el bucle.

Se invoca la Script a través del formulario **ejerc1-1.html**:

```
<html>
    <head>
      <title>Script 1-1</title>
    </head>
    <body>
      <form action="/cgi-bin/wxis.exe" method="post">
```

```
            <input type="hidden" name="IsisScript" value="ar/scrip1-1.xis">
            <input type=submit value="OK...">
        </form>
    </body>
</html>
```

El browser visualiza la siguiente pantalla:



A continuación se plantea el mismo ejemplo, pero no se especifica el path de la base sino que se determina a través de un archivo cipar. El mismo puede contener una línea por base en caso de ser necesario y para este caso, el mismo ha sido grabado con el nombre **cds.cip**.

Ej:

cds.*=c:\curso\cds.*

gente.*=c:\isis\data\gente\gente.*

Para una mejor visualización, los resultados son presentados en una tabla. La Script (**script1.xis**) queda codificada:

```
<IsisScript>
    <section>
    <display><pft>'Content-type: text/html'/#</pft></display>
    <parm name=cipar><pft>cat('ar/cds.cip')</pft></parm>
        <do task=mfnrange>
        <parm name=db>cds</parm>
        <parm name=count>10</parm>
            <loop>
                <display><pft>
                '<table border=1 width="100%">
                <tr><td rowspan="2" width="10%"> <b><center>', mfn(5)'
                  </center></b></td>
                <td>',v24,'</td>
                </tr>
                <tr><td>',(v70/),'</td></tr>
                </table><p>'
```

```
16.                     </pft></display>
17.                 </loop>
18.             </do>
19.         </section>
20. </IsisScript>
```

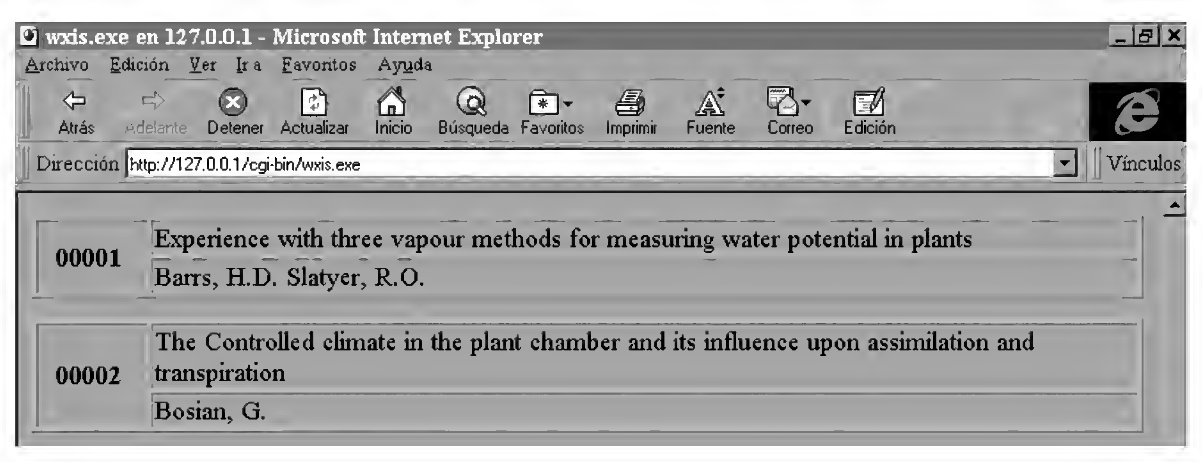
En el recuadro, se observa que la línea 4 contiene lo siguiente:

cat('ar/cds.cip')

La instrucción cat permite tomar como salida un archivo externo indicado por el formato (consultar manual de referencia del lenguage de formato). Para este caso, permite extraer del archivo **cds.cip** la ubicación de las bases de datos.

De la línea 10 a 15, se presenta un formato de tabla para visualizar los datos.

Mediante el browser se obtiene el mismo resultado pero visualizado en tabla:



Ejercicio Nº 2

A continuación se plantea un formulario para mostrar un rango de registros. En el mismo se solicita el nombre de la base de datos, el primer y último número de MFN a visualizar .La codificación del formulario (**ejerc2.html**) se plantea:

```
<html>
    <head>
    <title>Ejercicio Nº 2</title>
    </head>
    <body>
        <h1><center>Listado de Registros acotados</center></h1><p>
        <form action="/cgi-bin/wxis.exe" method="post">
            <input type="hidden" name="IsisScript" value="ar/script2.xis">
```

```
9.          Ingrese el nombre de la base: <input type="text" name="base"
            size="10"><br>
10.         Desde: <input type="text" name="from" size="2">
11.         Hasta: <input type="text" name="to" size="2"><br>
12.         <input type=submit value="Procesar">
13.      </form>
14.   </body>
15. </html>
```

En la línea 9 se solicita el nombre de la base y dicho valor se guarda en la variable *base*.

La línea 10 permite ingresar el número de MFN por el que se debe comenzar. Este número se almacena en la variable *from*.

La línea 11 requiere el último número de MFN a visualizar. En este caso, la variable que almacena este dato se denomina *to*.

El browser interpreta el archivo de la siguiente manera:



La Script que llama el formulario es ***script2.xis***, cuyo código es:

```
<IsisScript>
   <section>
     <display><pft>'Content-type: text/html'/#</pft></display>
     <field action=cgi tag=50>base</field>
     <field action=cgi tag=51>from</field>
     <field action=cgi tag=52>to</field>
     <parm name=cipar><pft>cat('ar/cds.cip')</pft></parm>
     <display><pft>'<h1><center>Registros recuperados</center></h1><p>'
     </pft></display>
     <do task=mfnrange>
        <parm name=db><pft>v50</pft></parm>
        <parm name=from><pft>v51</pft></parm>
        <parm name=to><pft>v52</pft></parm>
        <loop>
           <display><pft>
           '<table border=1 width="100%"><tr><td rowspan="2" width="10%">
           <b><center>',mfn(5)'</center></b></td><td>',v24,'</td></tr>
           <tr><td>',(v70/),'</td></tr> </table><p>'
```

```
16.             </pft></display>
17.          </loop>
18.       </do>
19.    </section>
20.</IsisScript>
```

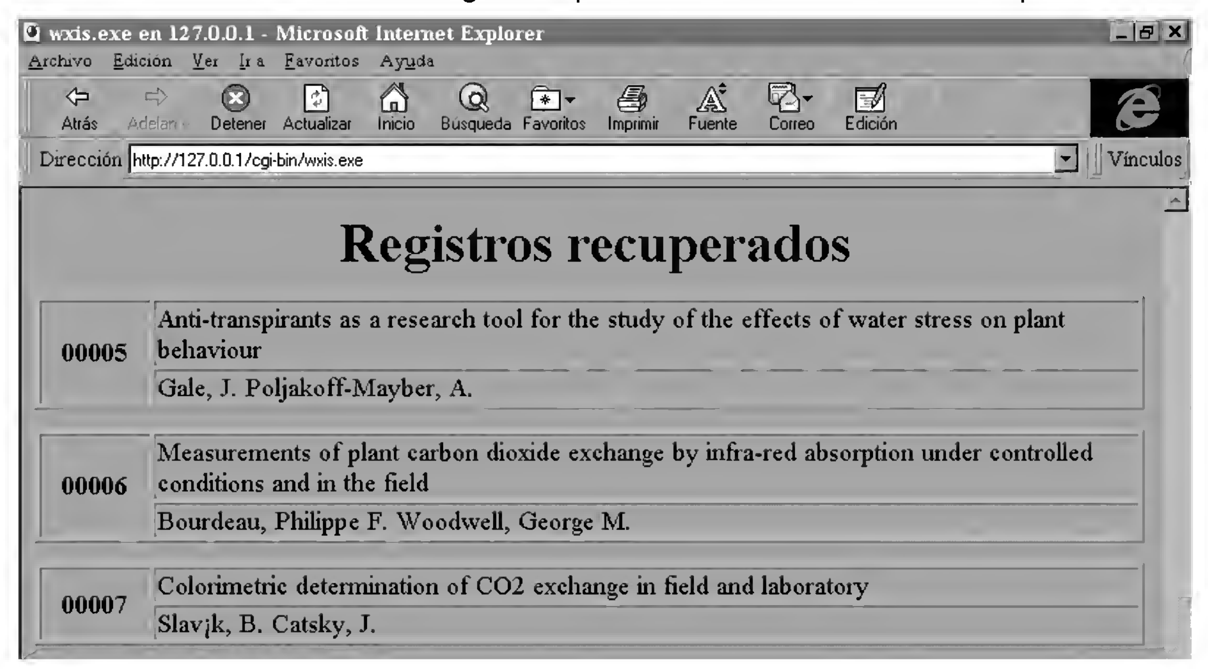
Las líneas 4, 5 y 6 reciben del formulario el nombre de la base de datos, el primer MFN y el último; y lo asignan a las etiquetas 50, 51 y 52 respectivamente. Este mecanismo se logra con la marcas *<field action="cgi" tag=nro_tag>* y *</field>* vistas en el ejercicio anterior.

En la línea 9 se plantea la tarea a realizar: mfnrange (sobre un rango de registros).

En la línea 10 mediante *<parm name=db>* se indica el nombre de la base de datos que para este caso está almacenado en la etiqueta 50. Lo mismo se hace para el primer y último número de MFN en las líneas 11 y 12 respectivamente. Esta información es interpretada por la tarea.

Dentro del *loop* a cada registro procesado se le aplica el formato con marcación html que se muestra en la línea 15.

El browser visualiza la siguiente pantalla como resultado de la operación:



Comentarios

Dentro de una Script se pueden incluir comentarios, los mismos deben estar encerrados entre las marcas *<!--* y -->. Por ejemplo:

<!-- este es un modelo de -->

Utilizando formatos de Isis

Si se desea utilizar, para visualizar los registros, los formatos que se diseñan en Isis, se tiene el inconveniente que los mismos no poseen marcación html y por lo tanto no existen saltos de línea en el resultado. Es por ello, que se puede invocar una instrucción que permite convertir los saltos de línea de Isis(/ y #) en una marca html. Esa instrucción es newline('marca_reemplazo') y se la debe incluir antes de invocar al formato (dentro del loop).

Ej. **<pft>newline('
'),@cds.pft</pft>** convierte los saltos de línea (/,#) del formato cds.pft, en marcas html
.

Variables del sistema.

En la versión WWWIsis, las variables del sistema (v1001 para Current, v1002 para Total) se agregaban automáticamente a cada uno de los registros. En WXIS se debe indicar los campos que se quieren incluir a los registros procesados. La función que lo permite es:

<field action=define tag=nro_de_tag>argumento</field>, donde *argumento* puede asumir los siguientes valores:

Isis_Current	Indice de la ejecución actual del <loop>.
Isis_Total	Total de veces posible para el <loop>.
Isis_From	Parámetro from del <loop>.
Isis_To	Parámetro to del <loop>.
Isis_Lock	Campo de almacenamiento del control de bloqueo de registros.
Isis_Status	Almacenamiento del estado de la ejecución de la tarea.
Isis_Item	Item de la lista.
Isis_Value	Cantidad de ítems calculados por el conteo de frecuencia.
Isis_Key	Llave actual.
Isis_Posting	Datos del posting actual.
Isis_Postings	Total de postings de la llave actual.
Isis_Items	Total de ítems en la lista.
Isis_ErrorInfo	Apuntador de error en la búsqueda.
Isis_Keys	Llaves para destacar texto.

Isis_MFN Número del campo de almacenamiento del MFN para exportación o importación de registros.

Isis_RecordStatus Número del campo que contendrá el estado del registro.

Manteniendo una correspondencia con WWWIsis, se podría asignar los mismos números a los tag, ej.: *<field action=define tag=1001>Isis_Current</field>*.

Nota: los valores de los argumentos solicitados son cargados en los campos indicados con *tag=*

A continuación se modifica el ejercicio anterior y se incorporan estos nuevos elementos.

Si el usuario ingresa un número mayor para el primer MFN respecto del último, la Script deberá validar esta situación y hacer un tratamiento de error. Para este caso, se contempla en la Script un salto, en la línea de ejecución, a otra sección de código que realiza el tratamiento del error.

El salto se programa con la marca *<flow action=jump>***nombre_del_rótulo***</flow>*. Cuando la Script llega a esta instrucción, se produce un salto en la ejecución del programa al segmento de código que comienza con el rótulo o etiqueta **nombre_del_rótulo**. En el sector donde se desea arribar con el salto, se indica la etiqueta con la marca *<label>***nombre_del_rótulo***</label>*.

Nota: Si el corte se produce dentro de la estructura <do> el salto debe encontrarse también dentro del </do>.

En la mayoría de los casos se desea que la ejecución del programa no ingrese a las líneas de código que le suceden al área a la que se arribó con el salto. Es decir, es necesario indicarle a la Script que abandone la corrida. Para cortar la ejecución, se utiliza la marca *<flow action=exit>***algo***</flow>*. Lo que se ha colocado entre las marcas *<flow....>* y *</flow>* no tiene efecto alguno en la Script; esto se hace porque entre las marcas siempre debe haber **algo**, de lo contrario la línea es ignorada.

Además, se han incorporado variables del sistema, como ser: el número del MFN que se está procesando (*Isis_Current*), el número total de registros (*Isis_Total*), el primer número de MFN con el cual se comienza el listado (*Isis_From*) y el último MFN que se debe alcanzar (*Isis_To*). Estos valores son asignados a etiquetas que luego se utilizan en el formato para, como por ejemplo, indicar el número total de registros, el nombre de la base de datos que se está procesando y que registro -respecto del total- se está visualizando.

Dada la longitud de las Script que se plantean, de ahora en más, los comentarios se encuentran dentro de la misma.

La Script (**scrip2-1.xis**) para este caso es la siguiente:

```
<IsisScript>
    <section>
    <display><pft>'Content-type: text/html'/#</pft></display>
    <field action=cgi tag=50>base</field>
    <field action=cgi tag=51>from</field>
    <field action=cgi tag=52>to</field>
    <parm name=cipar><pft>cat('ar/cds.cip')</pft></parm>

    <!-- la siguiente línea salta a la línea de código rotulada con el nombre
    ERROR, siempre y  cuando se valide la situación que v51 (valor del From)
    es superior a v52 (valor del To) -->
    <flow action=jump><pft>if val(v51) > val(v52) then 'ERROR' fi</pft></flow>

    <do task=mfnrange>
       <parm name=db><pft>v50</pft></parm>
       <parm name=from><pft>v51</pft></parm>
       <parm name=to><pft>v52</pft></parm>

    <!-- las líneas siguientes  cargan en las etiquetas 1001, 1002,1004 y 1005,
    los valores de Número actual del MFN, el límite del listado, el comienzo del
    mismo y el total de registros  respectivamente -->
       <field action=define tag=1001>Isis_Current</field>
       <field action=define tag=1002>Isis_To</field>
       <field action=define tag=1004>Isis_From</field>
       <field action=define tag=1005>Isis_Total</field>
       <display><pft>'<h1><center>Listado de Registros acotados</center></h1> <p>'</pft></display>

       <loop>
          <!--  con anterioridad se asignó a la etiqueta 50 el nombre de la base
          de datos, este valor no es visto dentro del loop (está fuera del Scope),
          por ello es que en la línea siguiente se lo importa dentro del Scope del
          Loop y se lo asigna a la etiqueta 1003 -->
            <field action=import tag=1003>50</field>
            <display><pft>,

          <!--  en la línea siguiente, si se está en el primer MFN se visualizan
          los datos del total de Registros y el nombre de la Base de datos. Para
          conocer si se está en el primer registro se pregunta si el registro
          actual (Isis_Current) es igual al Inicio (Isis_From) que fueron
          asignados a las etiquetas 1001 y 1004. El total de registros fue
          asignado a la etiqueta 1005 y el nombre de la base fue importado en
          la línea anterior en la etiqueta 1003 -->
               if v1001 = v1004 then '<h3>Total de Registros: ',v1005 ' ----- Base: ',v1003,'</h3><p>',fi,
```

```
22.          '<table border=1 width="100%" bgcolor="#808664"><tr><td
             rowspan="2" width="10%"><font color="#FFFFFF"><b><center>'
             ,mfn(5),'<br>'v1001,'/',v1002'</center></b></td><td><font
             color="#FFFFFF">',v24,'</td></tr><tr><td><font
             color="#FFFFFF">',(v70/),'</td></tr></table><p>'
23.        </pft></display>
24.     </loop>
25.   </do>

      <!-- La siguiente línea detiene la ejecución del script ya que no se produce
      el error. Entre las marcas <flow> debe existir alguna expresión, de lo
      contrario toda la línea es ignorada . -->
26.   <flow action=exit>0</flow>

      <!-- en las líneas siguientes se hace el tratamiento del error a partir de la
      designación del rótulo ERROR. -->
27.   <label>ERROR</label>
28.   <display><h3><i>ERROR !!!, los valores DESDE y HASTA est&aacute;n
      mal expresados</i></h3> <p></display>

29.   <flow action=exit>1</flow>

30.   </section>
31.</IsisScript>
```

Contemplando el requerimiento anterior (visualizar los registros del 5 al 7), el browser, visualiza la siguiente pantalla:



Si se hubieran invertido los valores (From=7 To=5), el browser –tras el tratamiento del error por parte de la Script- visualiza lo siguiente:



Controlando la ejecución de la Script.

Si se desea verificar la corrida de la script, se coloca al inicio de la misma las marcas *<trace></trace>*. La misma activa o desactiva el despliegue de la instrucción que está siendo ejecutada. Los modos posibles son normal(On), línea a línea(BR), o tabla(Table).

Si en el ejemplo de la Script se indica *<trace>BR</trace>*, el browser visualiza la siguiente pantalla:



Puede observarse en la figura el detalle de cada instrucción de la Script.

Búsquedas

En los ejemplos siguientes se plantean situaciones de búsquedas.

Ejercicio Nº 3

Comenzamos con un ejemplo sencillo de búsqueda. Se utiliza el formato de visualización visto en el ejercicio anterior.

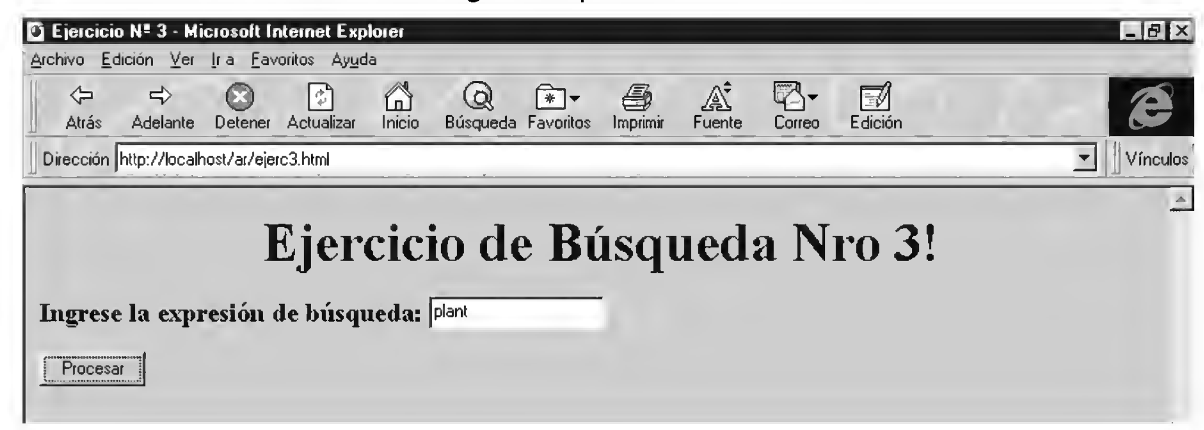
El formulario que se confecciona para solicitar la expresión de búsqueda es el siguiente:

```
1.  <html>
2.      <head>
3.          <title>Ejercicio Nº 3</title>
4.      </head>
5.      <body>
6.      <h1><center>Ejercicio de B&uacute;squeda Nro 3!</center></h1><p>
7.      <form action="/cgi-bin/wxis.exe" method="post">
8.          <input type="hidden" name="IsisScript" value="ar/script3.xis">
9.          <b>Ingrese la expresi&oacute;n de b&uacute;squeda: </b><input
            type="text" name="expbus" size="20"><br>
10.         <input type="hidden" name="path" value="c:\curso\"><br>
11.         <input type=submit value="Procesar">
12.     </form>
13.     </body>
14. </html>
```

En la línea 9 se solicita la expresión de búsqueda y se la pasa a la Script bajo la variable denominada *expbus*.

En la línea 10 se pasa la ubicación de la base de datos a través de un campo oculto denominado *path* cuyo valor es **c:\curso\\.**

El browser visualiza la siguiente pantalla:



La **script3.xis** tiene la siguiente codificación:

```
<IsisScript name=script3>
  <section>
     <display><pft>'Content-type: text/html'/#</pft></display>

     <!-- se asigna en la etiqueta 140 la expresión de búsqueda pasada a
     través de la variable expbus -->
     <field action=cgi tag=140>expbus</field>

     <!-- se asigna en la etiqueta 141 la ruta de la base de datos  -->
     <field action=cgi tag=141>path</field>

     <parm name=cipar><pft>
     'cds.*=',v141,'cds.*',/
     </pft></parm>

     <display><pft>'<h1><center>Registros recuperados</center></h1><h3>
     <center>Ejercicio Nro 3</center></h3><p>'</pft></display>

     <!-- la tarea que se va a realizar es búsqueda, es por ello  que  el  task
     es  search -->
     <do task=search>
     <parm name=db>cds</parm>

     <!-- a través del siguiente parm name se asigna la expresión de búsqueda
     que fue almacenada en la etiqueta 140 -->
     <parm name=expression><pft>v140</pft></parm>

     <field action=define tag=1001>Isis_Current</field>
     <field action=define tag=1002>Isis_Total</field>
     <field action=define tag=1009>Isis_ErrorInfo</field>

     <loop>
        <display><pft>
        '<table border=1 width="100%" bgcolor="#808664"><tr><td
        rowspan="2" width="10%"><font color="#FFFFFF"><b><center>'
        ,mfn(5),'<br>'v1001,'/',v1002'</center></b></td><td><font
        color="#FFFFFF">',v24,'</td></tr><tr><td><font
        color="#FFFFFF">',(v70/),'</td></tr></table><p>'
        </pft></display>
     </loop>

     <!-- en la siguiente línea se evalúa si se han encontrado registros. Ello se
     logra verificando el Total de Registros que se almacenó en la etiqueta
     1002 -->
     <display><pft>if v1002 = '0' then '<h3><i> No se encontraron registros
     para la expresi&oacute;n: ',v140,' </i></h3>' fi</pft></display>
     </do>
  </section>
</IsisScript>
```

El browser visualiza lo siguiente como resultado de la expresión de búsqueda **plant**:



Si se hubiera ingresado el término **planta**, se visualiza lo siguiente:



Nota: la tarea planteada en la Script es *<do task=search>*

En el ejercicio siguiente, se crea un formulario en el cual se puede elegir el formato de visualización con el que se quieren presentar los registros encontrados.

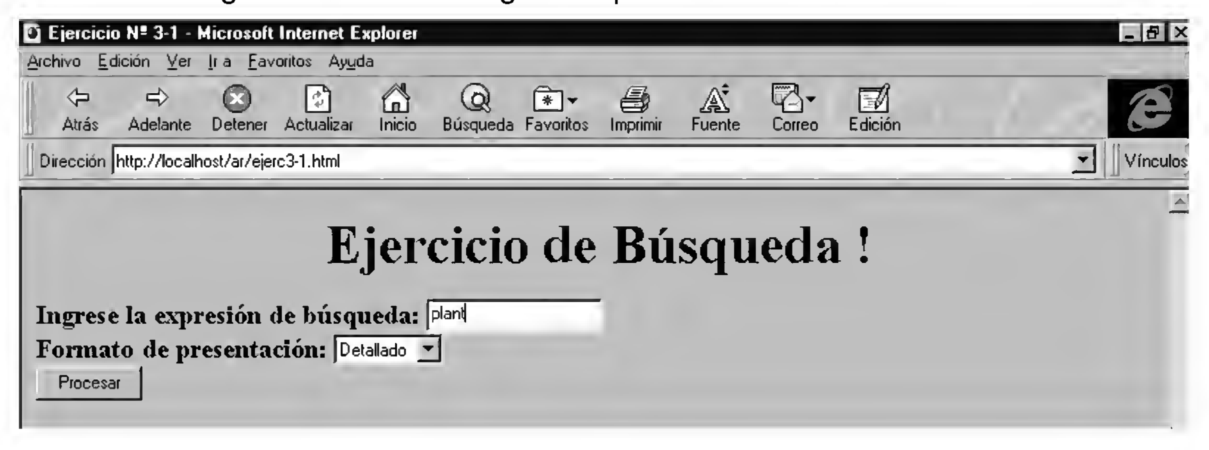
El formulario es el siguiente:

```
<html>
    <head>
    <title>Ejercicio Nº 3-1</title>
    </head>
    <body>
    <h1><center>Ejercicio de B&uacute;squeda !</center></h1><p>
```

```
        <form action="/cgi-bin/wxis.exe" method="post">
            <input type="hidden" name="IsisScript" value="ar/scrip3-1.xis">
            <b>Ingrese la expresi&oacute;n de b&uacute;squeda: </b><input
            type="text" name="expbus" size="20"><br>
            <b>Formato de presentaci&oacute;n: </b>
            <select name="format">
            <option value="1">Detallado</option>
            <option value="2">Corto</option>
            <option value="3">Titulo</option></select>
            <input type="hidden" name="path" value="c:\curso\"><br>
            <input type=submit value="Procesar">
        </form>
    </body>
</html>
```

A través de un select se le pasa a la Script la opción escogida por el usuario: Si eligió **Detallado** el valor asignado a la variable *format* es **1, 2** si su preferencia fue **Corto** y **3** si es por **Título.**

El navegador visualiza la siguiente pantalla:



La Script que invoca el formulario tiene la siguiente codificación (**scrip3-1.xis**):

```
<IsisScript name=scrip3-1>
    <section>
        <display><pft>'Content-type: text/html'/#</pft></display>

        <!-- se asigna a la etiqueta 140 el valor de la variable expbus pasada por
        el formulario que contiene la expresión de búsqueda -->
        <field action=cgi tag=140>expbus</field>

        <!-- se asigna a la etiqueta 141 el valor de la variable path pasada por el
        formulario que contiene la ruta de la base de datos -->
        <field action=cgi tag=141>path</field>

        <!-- se asigna a la etiqueta 142 el valor de la variable format  pasada por
        el formulario que contiene el formato elegido expresado en números (1,2
        o 3) -->
```

6. <field action=cgi tag=142>format</field>

```
      <!-- se indica un formato tipo cipar en la misma Script para indicar la
      ubicación de la base de datos y los formatos de visualización a utilizar.
      Notar que el formato3 corresponde a un formulario HTML. El valor de la
      etiqueta 141 fue pasado desde el formulario. -->
       <parm name=cipar><pft>
       'cds.*=',v141,'cds.*',/
       'formato1=',v141'cds.pft'/
       'formato2=',v141'cds1.pft'/
       'formato3=',v141'titulo.htm'/
       </pft></parm>

       <display><pft>'<h1><center>Registros recuperados</center></h1><h3>
       <center>Ejercicio Nro 3-1</center></h3><p>'</pft></display>

      <!-- la tarea a realizar es de búsqueda -->
       <do task=search>
         <parm name=db>cds</parm>
         <parm name=expression><pft>v140</pft></parm>
         <field action=define tag=1001>Isis_Current</field>
         <field action=define tag=1002>Isis_Total</field>
         <field action=define tag=1009>Isis_ErrorInfo</field>

         <loop>

           <!-- el formato elegido en el formulario y pasado a la Script no es
           visto dentro del loop (está fuera del Scope), es por ello que se
           necesita importarlo dentro del Scope del loop -->
           <field action=import tag=142>142</field>

           <display><pft>newline('<br>'),
           <!-- la siguiente línea evalúa que formato se debe tener en cuenta
           para la visualización de los registros, si es 1 o 2 directamente se
           invoca a los formatos respectivos. -->
           if v142 = '1' then @formato1 else if v142 = '2' then @formato2 fi fi,
           </pft></display>
           <!-- el formato 3 como se trata de un archivo en HTML, es
           necesario invocarlo con la instrucción cat. Pero antes de ello se
           debe indicar con la marca <htmlpft> que dicho formulario contiene
           un lenguaje de formato-->
           <display><htmlpft><pft>if v142 = '3' then cat('formato3') fi
           </pft></htmlpft></display>
         </loop>

         <!-- la siguiente línea evalúa si en la búsqueda se han encontrado
           registros. -->
         <display><pft>if v1002 = '0' then '<h3><i> No se encontraron registros
         para la expresi&oacute;n: ',v140,'</i></h3>' fi</pft></display>
         </do>
    </section>
</IsisScript>
```

Para visualizar los registros con el formato3 (Título) se emplea un archivo con marcas html (**titulo.htm**), que tiene además códigos de formato ISIS, que se identifican por estar entre las marcas *[pft]* y *[/pft]*. Todo lo incluido entre ellas será interpretado como funciones del lenguaje de formateo. Para que la script reconozca estas funciones se emplea *<htmlpft>* seguido de la función *cat('formato3')* que agrega línea a línea el contenido del archivo **titulo.htm** dentro de la script.

El archivo **titulo.htm** tiene la siguiente codificación:

```
1.  <html>
2.      <head>
3.          <title>Formato T&iacute;tulo</title>
4.      </head>
5.      <body>
6.          <h5>
7.          [pft]'<table border=1 width="100%" bgcolor="#808664">
            <tr>
                <td width="10%"><font color="#FFFFFF"><b><center>',mfn(5),'
                </center> </b></td>
                <td><font color="#FFFFFF">',v24,'</td>
            </tr>
            </table><p>
8.          '[/pft]
9.          <br></h5>
10.     </body>
11. </html>
```

Colocar entre las marcas **[pft][/pft]** los códigos html para que el formato aparezca en una tabla; estos códigos figuran como literales incondicionales (entre comillas simples).

Si se eligió el formato **Detallado**, el browser visualiza lo siguiente:



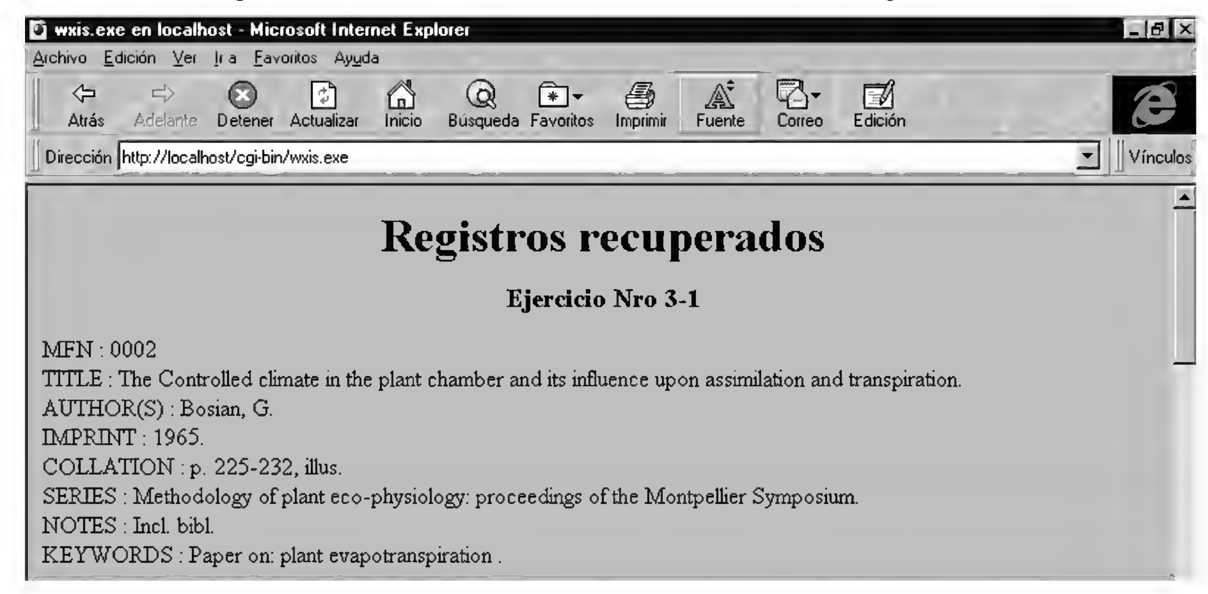
Registros recuperados

Ejercicio Nro 3-1

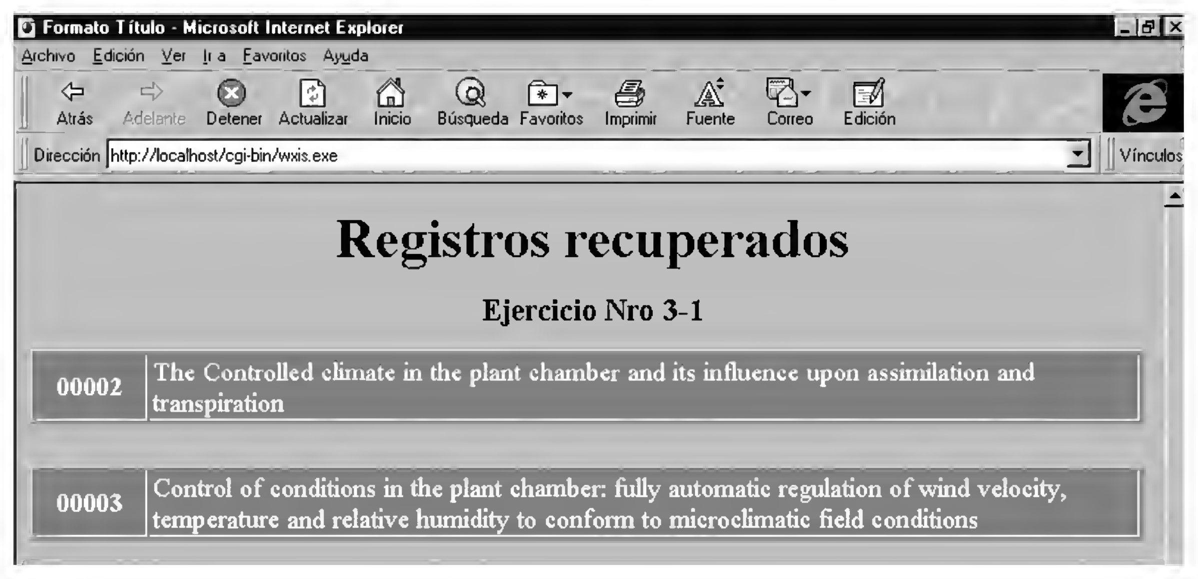
0002 - The Controlled climate in the plant chamber and its influence upon assimilation and transpiration. Bosian, G. 1965. p. 225-232, illus. (Methodology of plant eco-physiology: proceedings of the Montpellier Symposium)Incl. bibl.
KEYWORDS: Paper on: plant evapotranspiration .

0003 - Control of conditions in the plant chamber: fully automatic regulation of wind velocity, temperature and relative humidity to conform to microclimatic field conditions. Bosian, G. 1965. p. 233-238 , illus. (Methodology of plant eco-physiology: proceedings of the Montpellier Symposium)Incl. bibl.
KEYWORDS: Paper on: plant physiology moisture temperature wind measurement and instruments ecosystems .

Si se eligió el formato **Corto**, el browser visualiza lo siguiente:



Si se elige el formato **Título**, la pantalla que se despliega es la siguiente:



A continuación al mismo ejercicio se le incorpora un contador de autores, indicando para cada registro la cantidad existente; y luego un total al finalizar la visualización.

El esquema lógico de la script es el siguiente:

```
<do task=search>
        <parm name=.....>
        Definir campo cgi 9999 en cero
        <loop>
                Importar campo cgi 9999
                Reemplazar campo 9999 por (campo 9999 + nro. ocurrencias de
        v70)
                Mostrar datos del registro
                Exportar campo 9999
        </loop>
Mostrar 'Cant. autores: ', campo 9999
```

Notar que el contador de autores es un campo del **scope-section** definido fuera del **scope-registro**, que es importado e incrementado en cada ocurrencia del ciclo sumando la cantidad de autores; antes que termine el *<loop>* es nuevamente exportado al **scope-section** para que en el próximo ciclo sea importado sin perder la suma. De no exportarse el contador siempre será importado en cero.

El formulario que se envía es similar al anterior, es por ello que aquí se omite.

En la **scrip3-2.xis** se han resaltado las nuevas líneas agregadas; la codificación es la siguiente:

```
<IsisScript name=scrip3-2>
   <section>
      <display><pft>'Content-type: text/html'/#</pft></display>

      <!-- se asigna a la etiqueta 140 el valor de la variable expbus pasada por
      el formulario que contiene la expresión de búsqueda -->
      <field action=cgi tag=140>expbus</field>

      <!-- se asigna a la etiqueta 141 el valor de la variable path pasada por el
      formulario que contiene la ruta de la base de datos -->
      <field action=cgi tag=141>path</field>

      <!-- se asigna a la etiqueta 142 el valor de la variable format  pasada por
      el formulario que contiene el formato elegido expresado en números (1,2
      o 3) -->
      <field action=cgi tag=142>format</field>

      <!-- se indica un formato tipo cipar en la misma Script para indicar la
      ubicación de la base de datos y los formatos de visualización a utilizar.
      Notar que el formato3 corresponde a un formulario HTML. El valor de la
      etiqueta 141 fue pasado desde el formulario. -->
      <parm name=cipar><pft>
      'cds.*=',v141,'cds.*',/
      'formato1=',v141'cds.pft'/
      'formato2=',v141'cds1.pft'/
```

```
'formato3=',v141'titulo.htm'/
</pft></parm>

<display><pft>'<h1><center>Registros recuperados</center></h1><h3>
<center>Ejercicio Nro 3-1</center></h3><p>'</pft></display>

<!-- la tarea a realizar es de búsqueda -->
<do task=search>
<parm name=db>cds</parm>
<parm name=expression><pft>v140</pft></parm>
<field action=define tag=1001>Isis_Current</field>
<field action=define tag=1002>Isis_Total</field>
<field action=define tag=1009>Isis_ErrorInfo</field>

<!-- en la siguiente línea se “agrega” una nueva etiqueta a la Script que
servirá para el contador. Se le otorga una longitud de dos dígitos y se la
inicializa en cero. La acción es add (incorporar) -->
<field action=add tag=9999><pft>f(0,2,0)</pft></field>

<loop>

    <!-- el formato elegido en el formulario y pasado a la Script no es
    visto dentro del loop (está fuera del Scope), es por ello que se
    necesita importarlo dentro del Scope del loop -->
    <field action=import tag=142>142</field>

    <!-- dado que la etiqueta fue creada fuera del Scope, es necesario
    importarla dentro del loop y se le reasigna el mismo número -->
    <field action=import tag=9999>9999</field>

    <!-- la siguiente línea reemplaza el valor del campo 9999 dado que
    se actualiza el contador con la cantidad de autores del registro actual
    más el valor anterior del contador. Notar que la acción sobre el campos
    es replace -->
    <field action=replace tag=9999><pft>f(val(v9999)+nocc(v70),2,0)
    </pft></field>

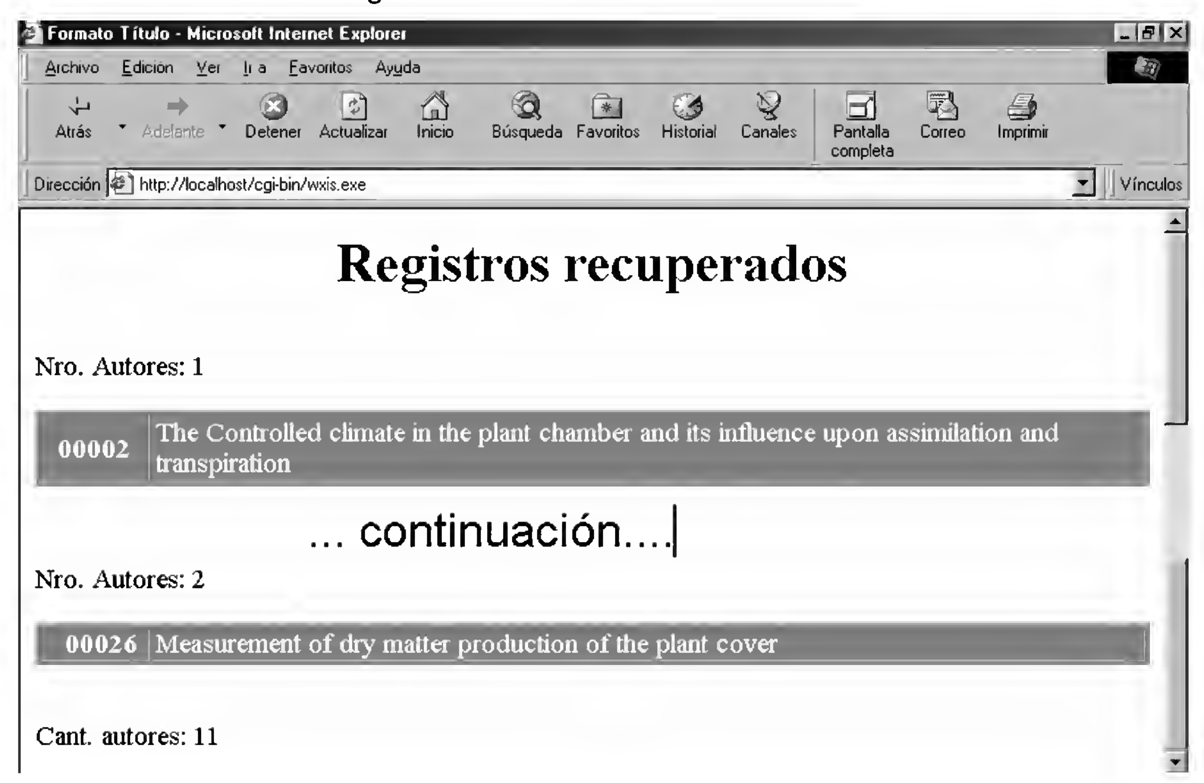
    <!-- la siguiente línea muestra el nro. de autores del registro -->
    <display><pft>'<br>Nro. Autores: ',f(nocc(v70),2,0),'<br>',

    <!-- la siguiente línea evalúa que formato se debe tener en cuenta
    para la visualización de los registros, si es 1 o 2 directamente se
    invoca a los formatos respectivos. -->
    newline('<br>'),
    if v142 = '1' then @formato1 else if v142 = '2' then @formato2 fi fi,
    </pft></display>
    <!-- el formato 3 como se trata de un archivo en HTML, es necesario
    invocarlo con la instrucción cat. Pero antes de ello se debe indicar
    con la marca <htmlpft> que dicho formulario contiene un lenguaje de
    formato-->
```

```
29.         <display><htmlpft><pft>if v142 = '3' then cat('formato3') fi
            </pft></htmlpft></display>
            <!-- antes que finalice el loop, es necesario exportar el valor del
            contador para poder visualizar el Total, dado que fuera del Scope
            este campo no se visualiza. -->
30.         <field action=export tag=9999>9999</field>
31.     </loop>

        <!-- la siguiente línea evalúa si en la búsqueda se han encontrado registros.
        Además se visualiza la cantidad de autores. -->
32.     <display><pft>'Cant. autores: ',v9999,if v1002 = '0' then '<h3><i> No se
        encontraron registros para la expresi&oacute;n: ',v140,'</i></h3>'
        fi</pft></display>
33.     </do>
34.   </section>
35.</IsisScript>
```

El browser visualiza el siguiente resultado:



Concepto Scope:

La estructura de una script define zonas en donde se alcanzan o se tiene acceso a las variables de ambiente WXIS y en otras, se acceden a las variables propias de los registros de la base de datos. Este concepto debe quedar bien

comprendido para programar una script; por esto es que se emplean las operaciones de *<field action=import...> y <field action=export...>* para pasar variables desde una zona (Scope) a otra. Las zonas en una script se pueden mencionar como **scope-registro** y **scope-section**.

La primera es la zona comprendida entre las marcas *<loop></loop>,* en donde se hace referencia a los campos del registro. Si dentro de este Scope se precisa una variable del **scope-section**, se la debe importar. Asimismo, si al terminar el *<loop>* se necesita procesar una variable que es parte del **scope-registro** se la debe exportar para poder tratarla fuera de esta zona. Más adelante se verá también que las marcas *<update></update>* corresponden a esta zona.

El **scope-section** es el que está por fuera del **scope-registro**, antes y después de las marcas *<loop></loop>*.

Visualización del Diccionario

A continuación se plantean ejercicios que permiten la visualización del Diccionario de Términos.

Ejercicio Nº 4

En el siguiente ejercicio se confecciona un formulario por el cual se ingresa el nombre de la base de datos, la primer palabra (o raíz) que se desea visualizar del diccionario y el límite de la visualización. La Script devuelve un página que muestra una lista de los términos del diccionario a partir de la palabra clave ingresada en **Desde**, y otorga la posibilidad de seleccionar del listado un término para buscar los registros relacionados.

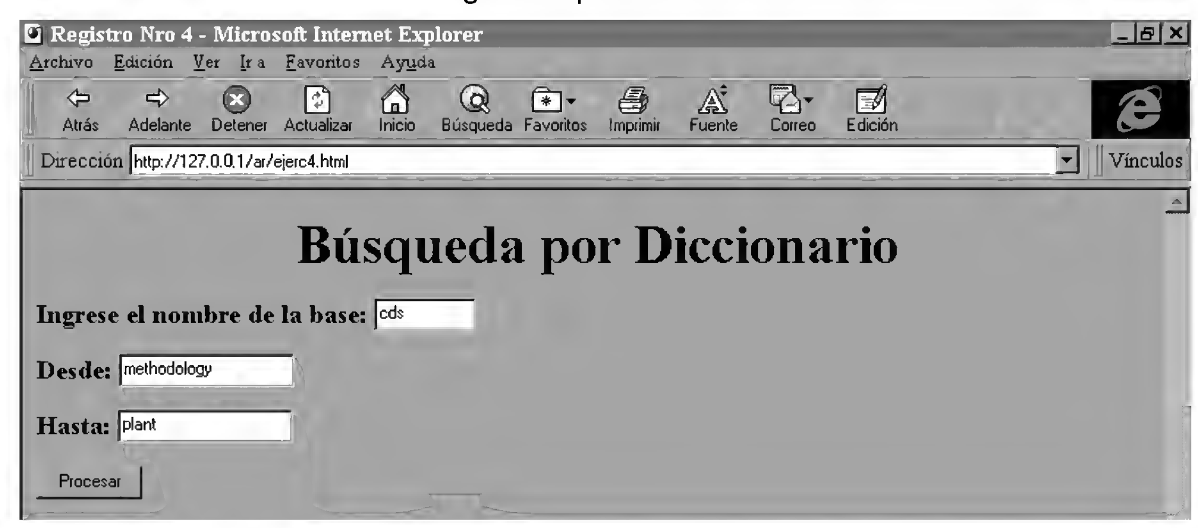
El **ejerc4.html** tiene la siguiente codificación:

```
<html>
    <head>
        <title>Ejercicio Nro 4</title>
    </head>
    <body>
    <h1><center>B&uacute;squeda por Diccionario</center></h1><p>
    <form action="/cgi-bin/wxis.exe" method="post">
        <input type="hidden" name="IsisScript" value="ar/script4.xis">
        <h4>Ingrese el nombre de la base: <input type="text" name="base"
        size="10"></h4>
        <h4>Desde: <input type="text" name="from" size="20"></h4>
        <h4>Hasta: <input type="text" name="to" size="20"></h4>
        <input type=submit value="Procesar">
    </form>
    </body>
```

```
15. </html>
```

Se puede observar que las líneas 9, 10 y 11 son similares a las que se han utilizado en ejercicio anteriores, pero en este caso lo que se va a ingresar en las variables *from* y *to* es texto.

El browser visualiza la siguiente pantalla:



Se solicita información de la base CDS y que la visualización del diccionario comience con el término **methogology** y finalice en **plant**.

La Script que invoca este formulario, por un lado visualiza el listado del diccionario, pero por otro, invoca a otra Script que retorna el resultado de la búsqueda. Es por ello, que se observa que dentro de la **script4.xis** se prepara el formulario para que llame a la **script5.xis**. La explicación de la Script se encuentra dentro de la misma como líneas de comentario. La codificación es la siguiente:

```
1.  <IsisScript>
2.     <section>
3.     <display><pft>'Content-type: text/html'/#</pft></display>
       <!-- en las líneas siguientes se asigna el nombre de la base al campo 50, el
       from al campo 51 y el to al campo 52 -->
4.     <field action=cgi tag=50>base</field>
5.     <field action=cgi tag=51>from</field>
6.     <field action=cgi tag=52>to</field>
7.     <parm name=cipar><pft>cat('ar/cds.cip')</pft></parm>

          <!-- La tarea es Keyrange, dado que se quiere visualizar un rango de los
          términos del diccionario -->
8.        <do task=keyrange>

9.           <parm name=db><pft>v50</pft></parm>
10.          <parm name=from><pft>v51</pft></parm>
11.          <parm name=to><pft>v52</pft></parm>
             <!-- si se desea que el diccionario sólo visualice 10 términos, entonces
             se deberá suprimir la línea anterior y reemplazarla por lo siguiente:
              <parm name=count>10</parm>    -->
```

```
<!-- en la línea siguiente se define una etiqueta que se utiliza para
almacenar la clave(término) actual que procesará el loop -->
<field action=define tag=1>Isis_Key</field>

<!-- se define una etiqueta para guardar la cantidad de posting de la
clave que procesará el loop -->
<field action=define tag=2>Isis_Postings</field>

<display><pft>'<h1><center>T&eacute;rminos del
Diccionario</center></h1><p> <hr><h4>Seleccione un t&eacute;rmino
y luego presione en Buscar</h4><p>'</pft></display>

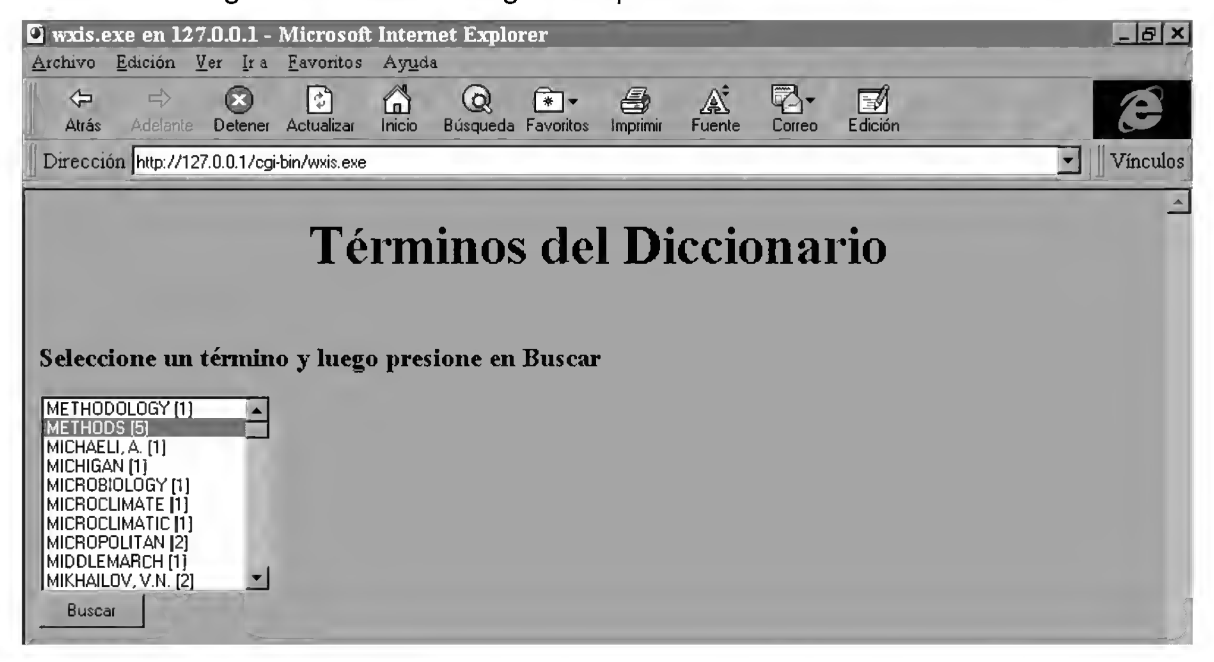
<!-- En las líneas siguientes preparo el formulario (con marcación html)
que va a visualizar el listado de los términos que voy a utilizar en la
búsqueda. Se invoca a la script5 que me muestra el resultado de la
búsqueda. La Script necesita contar con el nombre de la base de datos
y con la expresión de búsqueda. El nombre de la base se lo pasa a
través de un campo oculto denominado base y el valor es v50. El listado
de términos se visualiza dentro de un select y el término que seleccione
el usuario se almacena en un la variable claves. Este valor se debe
pasar a la script5 como término a buscar. -->
<display><pft>
'<form action="/cgi-bin/wxis.exe" method="post">',
'<input type="hidden" name="IsisScript" value="ar/script5.xis">',
'<input type="hidden" name=base value="',v50,'">',
'<select name="claves"size="10">',
</pft></display>

<!-- el siguiente loop colocará dentro del select, los términos del
diccionario -->
<loop>
      <display><pft>
      <!-- la información que se presenta es la clave (término) actual -
      v1-, y la cantidad de posting -v2-. Se presenta como option value
      porque es un elemento dentro del select, el cual puede ser elegido
      por el usuario -->
      '<option value="',v1,'">',v1,' [',v2,']</option>',
      </pft></display>
</loop>

<!-- cuando finaliza el loop se ha terminado de colocar los términos del
diccionario dentro del select, por lo tanto es necesario cerrar el select.
Luego se agrega el botón de submit para enviar el término
seleccionado por el usuario (guardado en la variable claves), y el
nombre de la base de datos (variable base). Y por último se cierra el
form -->
<display><pft>
'<br></select><input type=submit name=buscar value="Buscar">',
'</form>'
```

```
        </pft>
        </display>
    </do>
  </section>
</IsisScript>
```

El navegador visualiza la siguiente pantalla:



En la pantalla anterior, el usuario eligió el segundo ítem del select; cuando oprime en el botón buscar, se llama a la **script5.xis**, cuya codificación es la siguiente:

```
<IsisScript>
  <section>
  <display><pft>'Content-type: text/html'/#</pft></display>
  <!-- se recibe del formulario generado por la script4.xis, los valores
  correspondientes a la base de datos que se asigna al campo 50; y la clave
  para ejecutar la búsqueda que se la asigna al campo 51 -->
  <field action=cgi tag=50>base</field>
  <field action=cgi tag=51>claves</field>
  <parm name=cipar><pft>cat('ar/cds.cip')</pft></parm>

    <!-- como en los casos anteriores la tarea a realizar es Search-->
    <do task=search>
        <parm name=db><pft>v50</pft></parm>
        <parm name=expression><pft>v51</pft></parm>

        <display><pft>'<h1><center>Resultado de la
        B&uacute;squeda</center></h1> <p>'</pft></display>

        <!-- se visualiza el término elegido; el cual, ha sido motivo de la
        búsqueda -->
        <display><pft>'<p><h4>T&eacute;rmino buscado:
        ',v51,'</h4>'</pft></display>
```

```
        <loop>
           <display><pft>'<table border=1 width="100%"
           bgcolor="#808664"><tr><td rowspan="2" width="10%"><font
           color="#FFFFFF"><b><center>',mfn(5)'
           </center></b></td><td><font color="#FFFFFF">',v24,'</td></tr>
           <tr><td><font color="#FFFFFF">',(v70/),'</td></tr></table><p>'
           </pft></display>
        </loop>
     </do>
  </section>
</IsisScript>
```

La pantalla de visualización del resultado se omite presentarla, dado que es similar a los casos vistos.

En el ejemplo anterior, se ha tenido que confeccionar dos scripts para este procedimiento. ¿Se podría plantear el mismo ejercicio pero en una sola Script, es decir, unir la **script4.xis** y la **script5.xis**? Pues sí, y es lo que se desarrolla a continuación.

Se pretende confeccionar una Script que pueda evaluar qué formulario la invocó. Para ello, cada formulario deberá pasarle a la Script información a través de un campo oculto que le permita determinar qué debe procesar. A su vez, dentro de la Script, a través de las instrucciones de salto (*<flow action...>* se podrá controlar a qué sección debe pasar la ejecución).

Se ve primero el formulario (**ejerc4-1.html**) que solicita la información de la base y qué términos del diccionario visualizar. La codificación es similar al anterior y aquí se ha destacado en negrita la línea que se ha agregado:

```
1.  <html>
2.    <head>
3.       <title>Ejercicio Nro 4-1</title>
4.    </head>
5.    <body>
6.    <h1><center>B&uacute;squeda por Diccionario</center></h1><p>
7.       <form action="/cgi-bin/wxis.exe" method="post">
8.          <input type="hidden" name="IsisScript" value="ar/scrip4-1.xis">
9.          <h4>Ingrese el nombre de la base: <input type="text" name="base"
            size="10"></h4>
10.         <h4>Desde: <input type="text" name="from" size="20"></h4>
11.         <h4>Hasta: <input type="text" name="to" size="20"></h4>
12.         <input type="hidden" name=opcion value="DICCIONARIO">
13.         <input type=submit value="Procesar">
14.      </form>
15.   </body>
16. </html>
```

En la línea 12 se le pasa a la Script el campo oculto **opcion** con el valor **DICCIONARIO**.

La **scrip4-1.xis** que unifica a las **script4.xis** y **script5.xis**, tiene la siguiente codificación:

```
1.  <IsisScript>
2.     <section>
       <!-- esta Script une la Script4 y la Script5 -->

3.     <display><pft>'Content-type: text/html'/#</pft></display>

       <!-- se almacenan en las etiquetas 50, 51 y 52 los valores pasados por el
       formulario correspondiente al nombre de la base de datos, desde donde se
       visualiza el diccionario y el límite de visualización respectivamente -->
4.     <field action=cgi tag=50>base</field>
5.     <field action=cgi tag=51>from</field>
6.     <field action=cgi tag=52>to</field>

       <!-- se asigna a la etiqueta 53 la clave(término del diccionario) que el usuario
       eligió para la búsqueda-->
7.     <field action=cgi tag=53>claves</field>

       <!-- se asigna a la etiqueta 34 la variable opción que pasan los formularios.
       Opción va a valer DICCIONARIO si proviene del primer formulario o
       BUSCAR si viene del formulario que solicita la búsqueda del término-->
8.     <field action=cgi tag=34>opcion</field>

9.     <parm name=cipar><pft>cat('ar/cds.cip')</pft></parm>

       <!-- La próxima línea decide a qué parte de la Script se debe dirigir la
       ejecución. Este salto depende del valor de la etiqueta 34 (opción) pasada por
       los formularios. Dentro de la Script se tienen dos secciones: DICCIONARIO
       que corresponde a la Script4 y BUSCAR que corresponde a la Script5-->
10.    <flow action=jump><pft>,if S(v34):'DICCIONARIO' then 'DICCIONARIO' else
       'BUSCAR' fi,</pft></flow>

       <!--Aquí comienza el código de la script4. La explicación de cada línea
       se ha hecho en la script4.xis -->

11.    <label>DICCIONARIO</label>

12.       <do task=keyrange>
13.       <parm name=db><pft>v50</pft></parm>
14.       <parm name=from><pft>v51</pft></parm>
15.       <parm name=to><pft>v52</pft></parm>

16.       <field action=define tag=1>Isis_Key</field>
17.       <field action=define tag=2>Isis_Postings</field>
18.       <display><pft>'<h1><center>T&eacute;rminos del Diccionario</center>
          </h1><p> <hr><h4>Seleccione un t&eacute;rmino y luego presione en
          Buscar</h4><p>'</pft></display>
```

```
      <!-- Notar que aquí en vez de invocar a la script5.xis, se llama a si misma,
      es decir: scrip4-1.xis-->
      <display><pft>
      '<form action="/cgi-bin/wxis.exe" method="post">',/
          '<input type="hidden" name="IsisScript" value="ar/scrip4-1.xis">',/
          '<input type="hidden" name=base value="',v50,'">',/
          '<select name="claves"size="10">',/
          </pft></display>
          <loop>
             <display><pft>'<option value="',v1,'">',v1,' [',v2,']</option>',
             </pft></display>
          </loop>

          <!-- A diferencia de la script4.xis, aquí se envía al mismo formulario un
          campo oculto de nombre opcion con el valor BUSCAR, para que la
          próxima vez que ingrese a la Script, se dirija a la sección rotulada
          BUSCAR.-->
          <display><pft>
          '<br></select>',/
          '<input type="hidden" name=opcion value="BUSCAR">',/
          '<input type=submit value="Buscar">',/
      '</form>'</pft></display>
      </do>
   <flow action=exit>1</flow>
   <!-- aquí finaliza la Script4 -->

   <!-- aquí comienza la Script5. La explicación se dio en la script5.xis -->
   <label>BUSCAR</label>

   <do task=search>
   <parm name=db><pft>v50</pft></parm>
   <parm name=expression><pft>v53</pft></parm>
   <display><pft>'<h1><center>Resultado de la
   B&uacute;squeda</center></h1><p>'</pft></display>
   <display><pft>'<p><h4>T&eacute;rmino buscado:
   ',v53,'</h4>'</pft></display>

   <loop>
        <display><pft>'<table border=1 width="100%"
        bgcolor="#808664"><tr><td rowspan="2" width="10%"><font
        color="#FFFFFF"><b><center>',mfn(5)' </center></b></td><td><font
        color="#FFFFFF">',v24,'</td></tr><tr><td> <font
        color="#FFFFFF">',(v70/),'</td></tr></table><p>'          </pft></display>
   </loop>

   </do>
   <!-- aqui finaliza la Script5 -->

   </section>
</IsisScript>
```

Ordenamiento de registros en la visualización

Cuando el browser devuelve los registros como resultado de una búsqueda, los mismos son visualizados en forma ascendente por número de MFN. Si se desea invertir el orden de visualización, es decir, en forma descendente, se debe incluir en la Script (para este caso en la sección de la **scrip5.xis** antes del loop) la siguiente marca:

<parm name=reverse>On</parm>

Ahora se intenta obtener el resultado de la búsqueda ordenada o clasificada alfabéticamente por título.

Los retoques para este cometido se harán en la porción de código de la Script correspondiente a la visualización del resultado de la búsqueda: sección BUSCAR de la **script 4-1.xis**.

Antes de visualizar los registros encontrados, se debe cargar el resultado de la búsqueda en una lista. La misma se crea con un loop, con la siguiente marca:

<list action=load type=sort><pft>v24,'^n',mfn(0)</pft></list>

La acción es: cargar un lista tipo ordenada. Cada item de la misma contendrá el título propiamente dicho y un subcampo **n** que alojará el MFN del registro actual que se procesa dentro del loop. ¿Cómo opera? Dentro del loop carga la lista y la ordena por el campo 24 (título) y agrega el subcampo **n** que contiene el MFN de ese título, que permitirá referenciar al registro para obtener el resto de los datos necesarios en la visualización.

Luego de la carga de la lista se debe definir una nueva estructura con la tarea *task=list* de manera que se pueda obtener cada elemento de la lista (cargada previamente).

Antes de ello, se debe asignar cada elemento de la lista a un nuevo tag que será usado en el loop de visualización; para este caso se ha asignado el tag=1. Se logra con la siguiente instrucción:

<field action=define tag=1>Isis_Item</field>

Dentro del loop, para visualizar los registros, se utiliza un **ref([v1003]val(v1^n),formato)**. [Recordar que v1 corresponde a cada item de la lista ordenada.]

En la siguiente **scrip4-2.xis** únicamente se ha documentado lo nuevo:

```
<IsisScript>
  <section>
  <display><pft>'Content-type: text/html'/#</pft></display>
  <field action=cgi tag=50>base</field>
  <field action=cgi tag=51>from</field>
  <field action=cgi tag=52>to</field>
  <field action=cgi tag=53>claves</field>
  <field action=cgi tag=34>opcion</field>
  <parm name=cipar><pft>cat('ar/cds.cip')</pft></parm>
```

```
<flow action=jump><pft>,if S(v34):'DICCIONARIO' then 'DICCIONARIO' else
'BUSCAR' fi,</pft></flow>

<label>DICCIONARIO</label>
   <do task=keyrange>
   <parm name=db><pft>v50</pft></parm>
   <parm name=from><pft>v51</pft></parm>
   <parm name=to><pft>v52</pft></parm>
   <field action=define tag=1>Isis_Key</field>
   <field action=define tag=2>Isis_Postings</field>
   <display><pft>'<h1><center>T&eacute;rminos del Diccionario</center>
   </h1><p> <hr><h4>Seleccione un t&eacute;rmino y luego presione en
   Buscar</h4><p>'</pft></display>
   <display><pft>
   '<form action="/cgi-bin/wxis.exe" method="post">',/
       '<input type="hidden" name="IsisScript" value="ar/scrip4-1.xis">',/
       '<input type="hidden" name=base value="',v50,'">',/
       '<select name="claves"size="10">',/
       </pft></display>
       <loop>
          <display><pft>'<option value="',v1,'">',v1,' [',v2,']</option>',
          </pft></display>
       </loop>
       <display><pft>
       '<br></select>',/
       '<input type="hidden" name=opcion value="BUSCAR">',/
       '<input type=submit value="Buscar">',/
   '</form>'</pft></display>
   </do>
<flow action=exit>1</flow>
<label>BUSCAR</label>

   <do task=search>
   <parm name=db><pft>v50</pft></parm>
   <parm name=expression><pft>v53</pft></parm>
   <display><pft>'<h1><center>Resultado de la
   B&uacute;squeda</center></h1><p>'</pft></display>
   <display><pft>'<p><h4>T&eacute;rmino buscado:
   ',v53,'</h4>'</pft></display>

   <!-- la siguiente línea carga en una lista el campo 24 (título) y crea un
   subcampo n dentro del item de la lista para asignarle el número de MFN--
   >
      <loop>
          <list action=load
      type=sort><pft>mpl,v24,mpl,'^n',mfn(0)</pft></list>
      </loop>

      <!-- ahora que se tiene la lista cargada, se procede a la visualización.
      Para ello se crea una nueva tarea: list, dentro de la tarea search. -->
```

```
44.        <do task=list>
45.        <field action=define tag=1>Isis_Item</field>

                  <!-- dentro de un loop se hace el tratamiento de la lista para
                  visualizar los registros encontrados ordenados por título. -->
46.                <loop>
                       <!-- como se está dentro de otra tarea (otro Scope) es
                       necesario importar el nombre de la base de datos que se
                       utiliza para poder hacer el ref. -->
47.                    <field action=import tag=1003>50</field>

48.                    <display><pft>ref([v1003]val(v1^n),'<table border=1
                       width="100%" bgcolor="#808664"><tr><td rowspan="2"
                       width="10%"><font
                       color="#FFFFFF"><b><center>',mfn(5)'</center></b></td>
                       <td><font color="#FFFFFF">',v24,'</td></tr> <tr><td><font
                       color="#FFFFFF">',(v70/),'</td></tr></table><p>'),'<br>'
                       </pft></display>
49.               </loop>
50.        </do><!-- fin do task=list -->

51.     </do><!-- fin do task=search -->

52.  </section>
53.</IsisScript>
```

Notar que para poder referenciar cada registro a la base CDS, antes se debe importar al Scope la base mencionada mediante la instrucción *<field action=import tag=1003>50</field>,* que asocia a la misma al tag 1003 y es utilizada luego en la expresión **ref**.

Se tiene que recurrir a este método (de importación) dado que el proceso de visualización se genera a partir de un nuevo *<do>* anidado. Y dentro del mismo la base **cds** no es vista.

Actualización: Creación, Modificación y Baja de Registros

La actualización de las bases de datos a través de WXIS opera con bloqueo de registros de una manera muy sencilla. Esta metodología es necesaria en un ambiente de trabajo multiusuario y cliente/servidor, en el que se debe contemplar la posibilidad de que dos clientes pretendan acceder al mismo registro. Bajo esta circunstancia se debe permitir que solo uno tome posesión del registro, impidiendo que otros lo accedan hasta tanto no quede liberado por el primero. Este mecanismo es muy simple desde el punto de vista del programador, y más aún para la creación de nuevos registros que es totalmente transparente, ya que el mismo WXIS se encarga del bloqueo.

Para tomar posesión de un registro se debe indicar una llave que identifica al usuario y el tiempo máximo que dispone de la exclusividad del mismo. Si el usuario bloquea un registro y por cualquier eventualidad se interrumpe la comunicación, el registro quedará disponible luego que se cumpla este tiempo. Pero si el usuario retoma la comunicación y el registro continua bloqueado, wxis evalúa si la llave almacenada en el registro coincide con la del usuario para permitirle nuevamente tomar posesión del mismo.

Los datos del bloqueo (llave y tiempo de expiración) son almacenados en un nuevo campo que se agrega al registro: v1101 (en general), que en el subcampo **^i** guarda la llave y en **^t** el tiempo disponible para el bloqueo.

En el ejercicio que se verá, se emplea un formulario web para la carga de datos. La script programada permite altas, bajas y modificaciones considerando la estrategia de bloqueo indicada anteriormente.

Con WXIS se dispone de la información del ambiente del equipo que está haciendo la llamada al server. Existe una serie de variables que se pueden solicitar, por ej. Número de IP de la máquina del cliente, el navegador, etc. Con esta información se puede establecer la llave para identificar unívocamente al usuario; una opción podría ser IP+Fecha del sistema. Para el caso que se verá, el cliente ingresará la clave como si fuera un password.

Ejercicio Nº 5

A continuación se crea un formulario para recolectar datos y tomar algunas de las acciones siguientes: NUEVO (crear un nuevo registro), EDICION (modificar un registro) o BORRAR (eliminar un registro).

La codificación del formulario es la siguiente:

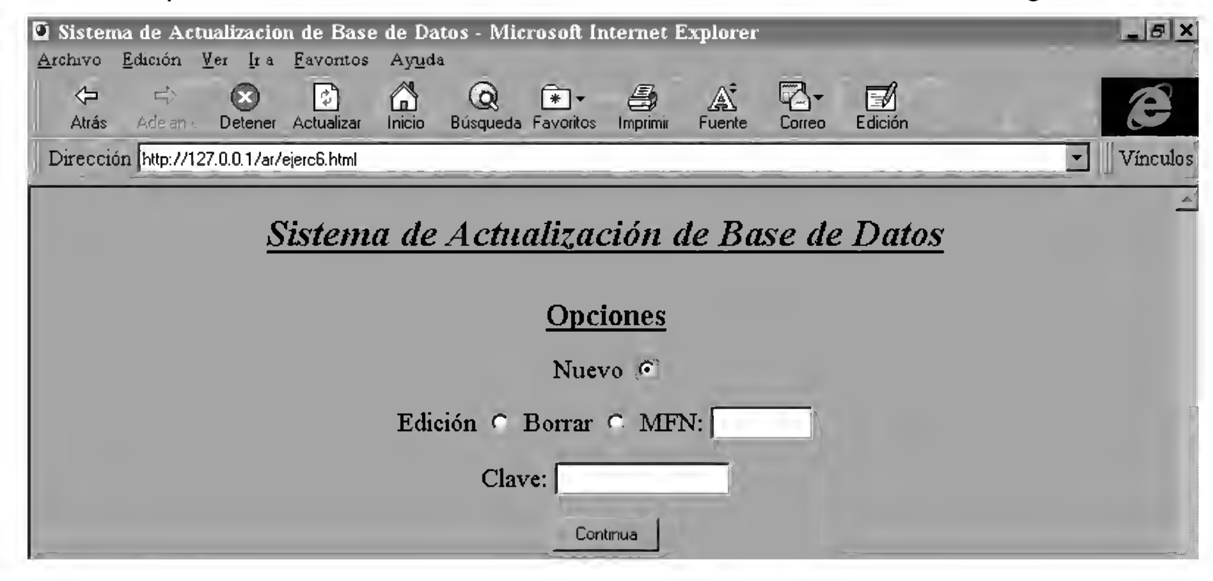
```
<html>
    <head>
        <title>Sistema de Actualización de Base de Datos</title>
    </head>
    <p align="center"><font size="5"><em><strong><u>Sistema de
    Actualizaci&oacute;n de Base de Datos</u></strong></em></font></p>
        <form action="/cgi-bin/wxis.exe" method="post">
            <input type="hidden" name="IsisScript" value="ar/script6.xis">
```

```
9.              <center>
10.             <p><font size="4"><strong><u>Opciones</u></strong></font></p>
11.             <p>Nuevo <input type="radio" checked name="opcion"
                value="NUEVO"></p>
12.             <p>Edición <input type="radio" name="opcion"
                value="EDICION">

13.             Borrar <input type="radio" name="opcion" value="BORRAR">
14.             MFN: <input type=text name=id size=10></p>
15.             <p>Clave: <input type="password" size="20" name="clave"></p>
16.             <p><input type="submit" value="Continua"> </p>
17.             </center>
18.         </form>
19.     </body>
20.</html>
```

A través de las líneas 11, 12 y 13 se indica cual es la acción que se desea realizar. Si se indica EDICIÓN o BORRAR, se deberá ingresar el MFN. A través de este formulario se invoca a la **script6.xis** que toma alguna de las tres acciones que aquí se proponen dependiendo del valor cargado en la variable **opcion**. Si el usuario eligió Nuevo, entonces opcion=NUEVO, si prefirio Edición, opcion=EDICION y opcion=BORRAR para el caso que haya seleccionado Borrar. Este variable **opcion**, más la clave y el mfn, son pasados a la Script.

La pantalla de visualización en el browser del formulario es la siguiente:



Seguimiento lógico del ejercicio:

- A partir de **ejerc6.htm** se llama al **Script6.xis**
- Inicio de **Script6.xis**
  Según **opcion** hacer:
  - ALTA
    Presentación de **formulario.htm**
  - EDICION
    Presentación de **formulario.htm**
    Se carga la llave y mfn
    Si el registro está disponible
    se bloquea el registro
    Sino
    'Error !'
  - BAJA
    Presentación de **formulario.htm**
    Se carga la llave y mfn
    Si el registro está disponible
    se bloquea el registro
    Sino
    'Error !'

En la presentación de **formulario.htm** se cargan o muestran los datos del registro (Título, Autor y Notas). Esta página web volverá a llamar a la **script6.xis** que también tendrá definida una variable 'oculta' **opcion** que valdrá:

OK_ALTA: Si la opción anterior fue ALTA

OK_EDICION: Si la opción anterior fue EDICION

OK_BAJA: Si la opción anterior fue BAJA

La llave y el mfn también son guardados en campos ocultos

  - OK_ALTA
    Actualización de la base de datos
  - OK_EDICION
    Se carga la llave y mfn
    Se actualiza la base
    Se evalúa Si el código de error = 0
    'Operación exitosa'
    Sino
    'Error, no se pudo actualizar el registro'
  - OK_BAJA
    Se carga la llave y mfn
    Se actualiza la base
    Se evalúa Si el código de error = 0
    'Operación exitosa'
    Sino
    'Error, no se pudo actualizar el registro'

Creación de registros (NUEVO)

La **script6.xis** recibe el valor **opcion** y dependiendo de él, deriva la ejecución de la Script a distintas zonas codificadas para tal fin. En primer término se verá el código correspondiente para el tratamiento de la opción **NUEVO**:

```
<IsisScript>
    <section>
    <display><pft>'Content-type: text/html'/#</pft></display>

    <!-- en la próxima línea se guarda el nombre de la Script-->
    <field action=cgi tag=5000>IsisScript</field>

    <!-- en la próxima línea se guarda la opcion que se pasa del formulario-->
    <field action=cgi tag=5001>opcion</field>

    <!-- en la próxima línea se guarda la llave pasada por el formulario-->
    <field action=cgi tag=5002>clave</field>

    <!-- en la próxima línea se guarda el numero de MFN pasado por el
    formulario-->
    <field action=cgi tag=6001>id</field>

    <!-- en la próxima línea se crea una nueva variable para indicar el lugar de
    la base de datos-->
    <field action=add tag=3030><pft>'c:\curso\'</pft></field>

    <!-- se indica el cipar para los caminos de las bases de datos-->
    <parm name=cipar>
    <pft>'cds.*='v3030'cds.*'/</pft></parm>

    <!-- se evalúa el valor de opcion y dependiendo de él se hace un salto dentro de
    la Script a los apartados NUEVO, EDICION o BORRAR. Cuando la script es
    invocada desde forumulario.htm, la variable opcion contiene OK_NUEVO,
    OK_EDICION u OK_BORRAR que son las secciones de la script que terminan
    de procesar la opción -->
    <flow action=jump><pft>v5001</pft></flow>

    <!--Aqui comienza la opcion NUEVO -->
    <label>NUEVO</label>

    <!-- lo único que se debe realizar en NUEVO es pasar el control al
    formulario que se encarga de colectar los datos para luego grabarlo en la
    base. Ese formulario se denomina formulario.htm. El mismo cuenta con
    lenguaje de formateo en su interior, es por ello que se han agregado las
    marcas htmlpft y se lo habilita a través de la instrucción cat.
    Posteriormente el formulario invocará nuevamente a la script6.xis para
    realizar el tratamiento de OK_NUEVO-->
```

```
<display><htmlpft><pft>cat('ar/formulario.html')</pft></htmlpft></display>
<flow action=exit>1</flow>
<!-- aqui finaliza la opcion NUEVO -->
```

El archivo **formulario.htm** presenta un formulario para el ingreso de datos (Título, Autor y Notas). Así mismo es empleado para las opciones de EDICION y BORRAR, donde mostrará el contenido de estos campos.

El browser visualiza la siguiente pantalla:



Este formulario vuelve a llamar a la **script6.xis**. Notar en el código de la página que se vuelve a definir la variable **opcion** que contendrá OK_NUEVO, OK_EDICION y OK_BORRAR para terminar de procesar la opción.

El código del mismo es:

```
<html>
    <head>
        <title>Nuevo Registro</title>
    </head>
        <form action="/cgi-bin/wxis.exe/">
            <input type="hidden" name="IsisScript" value="ar/script6.xis">
            <!-- es necesario pasarle nuevamente  a la Script la llave y el número
            de MFN. Esos datos se pasan a través de campos ocultos. Para el
            caso de Nuevo, el valor correspondiente a la etiqueta 6001 es nulo-->
            <input type="hidden" name="clave" value="[pft]v5002[/pft]">
            <input type="hidden" name="id" value="[pft]v6001[/pft]">

            <!-- Aquí el valor de opcion depende del valor de la etiqueta 5001
            definida en la script6.xis y ese mismo valor indicará el flujo en la
            ejecución de la script6.xis cuando vuelva a ser ejecutada.
            IMPORTANTE: notar como son asignados los valores para el
            campo oculto opcion. Tener presente las Comillas Dobles!!! son
            parte de las literales -->
            <input type="hidden" name="opcion" value=
            [pft]if v5001 = 'NUEVO' then '"OK_NUEVO"' fi[/pft]
            [pft]if v5001 = 'BORRAR' then '"OK_BORRAR"' fi[/pft]
            [pft]if v5001 = 'EDICION' then '"OK_EDICION"' fi[/pft]>
```

```
        <!-- El título que se visualice en el formulario también depende del
        valor de la etiqueta 5001 -->
        <p align="center"><font size="5"><em><strong><u>
        [pft]if v5001 = 'NUEVO' then 'Nuevo Registro' fi[/pft]
        [pft]if v5001 = 'BORRAR' then 'Eliminaci&oacute;n del Registro' fi[/pft]
        [pft]if v5001 = 'EDICION' then 'Edici&oacute;n de Registro' fi[/pft]
        </u></strong></em></font></p>

        <!--A continuación se presenta una entrada de texto para que se
        ingrese el Título; notar que este valor muestra el contenido de la
        etiqueta 24 del registro, que para opcion=NUEVO, estará vacío, pero
        para EDICION y BORRAR mostrará el valor del campo.-->
        Titulo:   <input type="text" name="titulo" size=50 value="
        [pft]v24[/pft]"> <br>

        <!-- Lo mismo se indica para el caso del autor. Aquí se utiliza un
        elemento textarea para poder ingresar en cada línea un autor, así
        también mostrará el campo v70 para las opciones EDICION y
        BORRADO notar que se ha encerrado entre paréntesis la etiqueta v70
        para indicar que es un campo repetible; y de haber más de un autor,
        los mismos se visualizan uno por renglón..-->
        Autor:   <textarea name="autor" cols="50"
        rows="3">[pft](v70/)[/pft]</textarea><br>

        <!-- se solicita también, un campo para notas.-->
        Notas:  <input type="text" name="notas" size=50
        value="[pft]v50[/pft]"><br>

        <!-- NOTA: La presencia de las etiquetas v24, v70 y v50 son para
        visualizar los valores de dichos campos para el caso de EDICION y
        BORRAR, para NUEVO estos campos están vacíos. El submit llama
        nuevamente a la script6.xis para terminar la opción-->
        <input type="submit" value="OK">
      </form>
   </body>
</html>
```

Al salir del **formulario.htm**, se ejecuta nuevamente la **script6.xis** que trae el valor OK_NUEVO dentro de la variable **opcion**. Es por ello que la parte de código de la script que se ejecuta es la rotulada con OK_NUEVO y que corresponde al grabado en disco del nuevo registro. El código de esa porción de la script es la siguiente:

```
   <!--Aqui comienza la opcion OK_NUEVO  en la script6.xis-->
   <label>OK_NUEVO</label>

        <!—se indica la tarea para cargar el nuevo registro, la misma es: . -->
        <do task=update>
```

```
<!—se indica la base en la cual se carga el nuevo registro -->
<parm name=db>cds</parm>


<!-- se indica New como parámetro de mfn ya que se trata de un
nuevo registro -->
<parm name=mfn>New</parm>

<!-- se explicita la fst que se utiliza para invertir el registro. La misma
será utilizada al momento de grabar el registro y de esta forma se
actualiza el archivo invertido. Se podría haber indicado las líneas de la
fst. Para este caso como ya se cuenta con el archivo cds.fst, se lo
carga con la instrucción cat. -->
<parm name=fst><pft>cat('cds.fst')</pft></parm>

<!-- se define un campo para almacenar el estado del registro en la
base de datos( información que brinda el Isis_Status). Este dato sirve
luego para chequear si la operación de grabación fue exitosa. -->
<field action=define tag=1102>Isis_Status</field>

<!-- se indica el proceso de actualización en la base de datos. -->
<update>
   <!-- todo lo que figure dentro del update se va a grabar en la base
   indicada y en el nuevo MFN. Se necesita traer los datos que se
   cargaron en el formulario y que se encuentran en el cgi. Si se los
   lee dentro del update, no se los necesita importar. También se los
   podría traer fuera del scope-update pero deberían ser importados. -
   ->

   <!-- Se carga en el tag 24 el valor pasado por el formulario en el
   campo titulo -->
   <field action=cgi tag=24>titulo</field>

   <!-- Se carga en el tag 70 el valor pasado por el formulario en el
   campo autor -->
   <field action=cgi tag=70>autor</field>

   <!—En el textarea del formulario se cargó en líneas distintas un
   autor diferente. Si se intenta almacenar la información tal cual se
   pasa del formulario, en la base de datos  todos los autores
   cargadas se encontrarán en una sola ocurrencia. Para evitar esto
   es necesario indicarle que cada Enter (CR) debe ser tratado como
   una nueva ocurrencia dentro del campo autor. Es decir, se debe
   tratar la repetibilidad del campo autor antes de almacenarlo. Para
   ello se reemplaza lo del campo 70 por un formato,  indicando un
   nuevo argumento en el field con el parámetro split, que le indica
   que cada vez que encuentre un CR, lo reemplace por una nueva
   ocurrencia. -->
   <field action=replace tag=70 split=occ><pft>(v70/)</pft></field>

   <!-- se  almacena el campo notas -->
```

```
            <field action=cgi tag=50>notas</field>

            <!-- Como el registro es nuevo, no es necesario pasar la llave ya
            que el bloqueo lo maneja internamente wxis. Si se debe indicar
            cuál va a ser el estado del registro después de grabar. Las posibles
            opciones son: unlock (desbloquear), lock (grabe y bloquee) y delete
            (borra, porque la acción de delete es grabar). Para este caso se
            utiliza Unlock; si la opción fuera Lock, en este caso se debe indicar
            la llave y el tiempo de  expiración del bloqueo. Esta acción se
            indica con la marca <write>.
            Al momento de grabar también se actualizó el archivo invertido -->
            <write>Unlock</write>

            <!--  en la línea siguiente, se invoca con la instrucción cat, al
            archivo resultados.html que visualiza el resultado de la operación
            dependiendo del valor de la variable 1102 que corresponde al
            Isis_Status.  Debe encontrarse antes que se cierre el Update,
            porque de lo contrario estaría fuera del scope y por lo tanto no
            sería vista. -->
            <display><htmlpft><pft>cat('ar/resultados.html')</pft></htmlpft></di
            splay>
        </update>
        </do>
        <!--  para que no ingrese en la otra sección de código, es necesario
        finalizar la script en este punto. -->
        <flow action=exit>1</flow>
    <!-- aqui finaliza la opcion OK_NUEVO -->
```

IMPORTANTE: como se observa, la script para OK_NUEVO no contiene un LOOP. Esto se debe a que la operación es sobre un solo registro. El Loop está pensado más bien, para una búsqueda, un keyrange, etc.

La página **resultado.html** contiene la siguiente codificación:

```
<html>
    <head>
        <title>Resultado</title>
    </head>

    <body>
    [pft],If val(v1102) <> 0 then 'Error: ',v1102,'   Ha habido problemas con la
    operaci&oacute;n...' else '<b><strong>Operaci&oacute;n exitosa
    !!!</strong></b><br>' fi[/pft]

    <a href="/ar/ejerc6.html">Volver...</a>
    </body>
```

```
9. </html>
```

El formulario hace un tratamiento del estado de la operación analizando el valor de la variable 1102 (*Isis_Status*). Si el valor es cero (0) la operación fue exitosa, es decir, no hubo problemas para la grabación del registro. Caso contrario, de no completarse la misma se muestra el código de error. Además se contempla un retorno al formulario **ejerc6.html.**

Para este caso, si la operación fue exitosa, se visualiza la siguiente pantalla:



Modificación de Registros (EDICION)

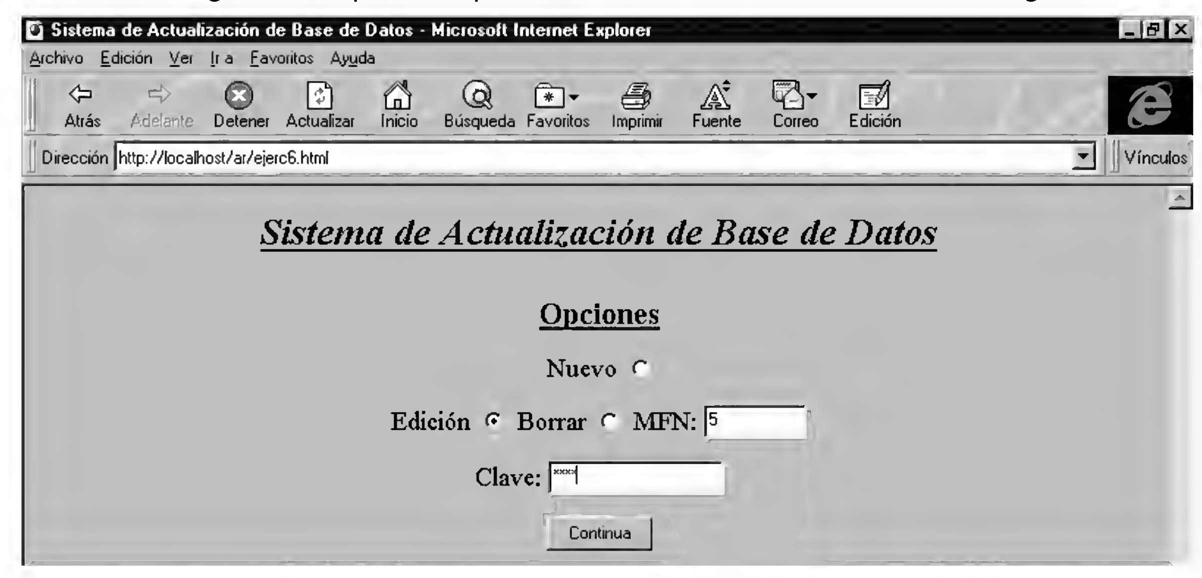
Se aborda ahora el modelo de Edición de un Registro que ya ha sido cargado.

Primero se verifica si el registro está bloqueado. De estarlo se informa; de lo contrario se envía un formulario que presenta la información que tiene cargado el registro con la posibilidad de modificarlo.

Para este caso, se trabaja con la misma script: **script6.xis.**

Para modificar un registro se debe indicar la base de datos, el número de mfn que se modifica y la llave del cliente.

El formulario que se utiliza es el que se ha venido utilizando para la carga de nuevos registros; la pantalla para este caso de modificación es la siguiente:



Como se observa en el gráfico, se ha indicado la edición del registro número 5 y también se ha colocado la clave (llave).

Cuando se hace click en el botón **Continúa**, se pasa al control a la **script6.xis**. En este caso, el formulario pasa el valor **EDICION** a través de la variable oculta **opcion.** Recordar que la script analiza el valor de la opción y

dependiendo del mismo deriva el control a una sección dentro de la script. Esa porción de código que realiza el desvío del flujo de la ejecución es:

```
<flow action=jump><pft>v5001</pft></flow>
```

donde v5001 es el valor de **opcion**. Como dicha variable posee el valor EDICION, entonces la script realiza el salto al rótulo con este nombre.

En la Script se debe indicar el tiempo que dispone el usuario para modificar el registro. Ello se logra con la marca

<parm name=expire>tiempo_en_segundos</parm>. Si el tiempo expira, el registro se desbloquea para ese usuario, con lo cual, ya no tendrá la posibilidad de modificarlo si otro usuario ha tomado posesión del mismo.

Su código es:

```
<!--Aqui comienza la opcion EDICION -->
<label>EDICION</label>

<!--la tarea que se realiza es de actualización, por ello update -->
<do task=update>

    <!--se indica la base, el número de mfn (tag 6001) y la llave (tag 5002)
    -->
    <parm name=db>cds</parm>
    <parm name=mfn><pft>v6001</pft></parm>
    <parm name=lockid><pft>v5002</pft></parm>

    <!-- se indica el tiempo total que dispone el cliente  para actualizar el
    registro (60 segundos) -->
    <parm name=expire>60</parm>


    <!-- se define el  tag para verificar el estado de registro -->
    <field action=define tag=1102>Isis_Status</field>

    <!-- se define un nuevo tag 1101 para dejar grabado en el registro la
    información del bloqueo. Notar que este campo no es  importado al
    scope-update ya que wxis los agrega como parte del registro. -->
    <field action=define tag=1101>Isis_Lock</field>

    <update>

        <!-- en la próxima instrucción se intenta bloquear el registro con la
        llave que se posee. De lograrse, el registro estará disponible por 60
        segundos para ser modificado por el cliente. -->

    <write>Lock</write>

        <!-- es necesario importar la variable opcion, que esta en el tag
        5001 para que se vea; y luego poder pasarla al formulario -->
        <field action=import tag=5001>5001</field>
```

```
            <!-- también es necesario importar la clave, para pasarla luego al
            formulario -->
            <field action=import tag=5002>5002</field>

            <!-- lo mismo para el MFN para luego pasarlo al formulario -->
            <field action=import tag=6001>6001</field>

            <!-- se indaga si se ha bloqueado el registro a través del Isis_Status
            que corresponde a la etiqueta 1102 (val(v1102) <> 0); y de estar
            bloqueado por otro usuario, se emite una leyenda de error -->
            <display><pft>If val(v1102) <> 0 then '<h3><b>Error: ',v1102,'   El
            registro no est&aacute; disponible...</b></h3>' fi</pft></display>

            <!-- como el registro no esta bloqueado (val(v1102) = 0), se llama al
            formulario que presentará la información del registro con la opción
            de modificar sus datos-->
            <display><htmlpft><pft>if val(v1102) = 0 then
            cat('ar/formulario.html') fi</pft></htmlpft></display>

        </update>
    </do>
    <flow action=exit>1</flow>
<!-- aqui finaliza la opcion EDICION -->
```

A **formulario.html** se lo llama pasándole el nombre de la base de datos, el mfn que se actualiza, la llave del usuario (estos datos son necesarios pasarlos porque luego los necesitará la **script6.xis** para proceder a la grabación del registro) y el valor EDICION en **opcion.**

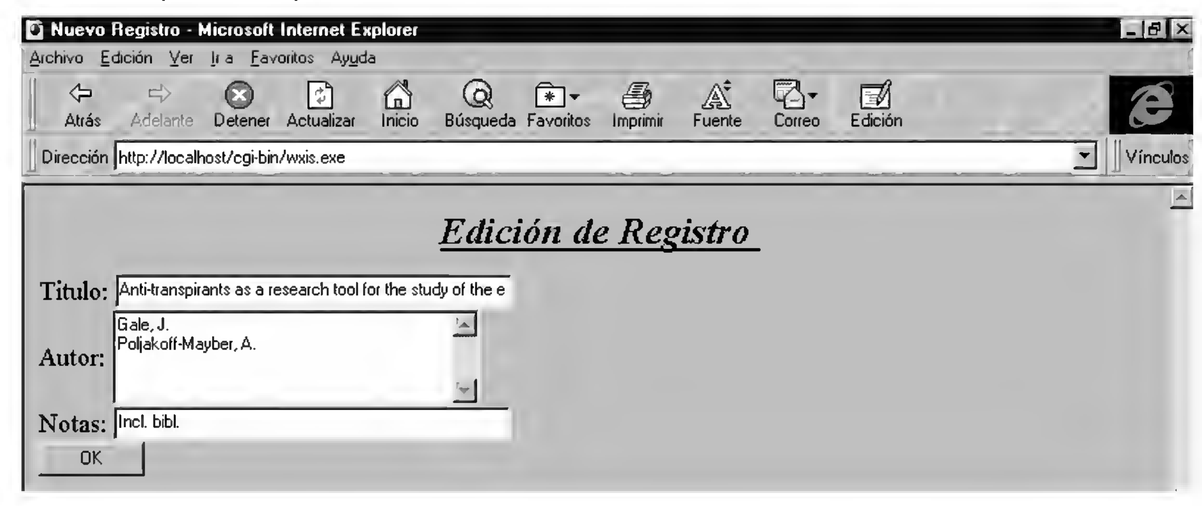
El formulario mostrará el registro y aceptará las modificaciones, luego indaga el valor de **opcion** y dependiendo de él, realiza la actualización de la variable y también presenta el título en función a su contenido:

```
<input type="hidden" name="opcion" value=
   [pft]if v5001 = 'NUEVO' then '"OK_NUEVO"' fi[/pft]
   [pft]if v5001 = 'BORRAR' then '"OK_BORRAR"' fi[/pft]
   [pft]if v5001 = 'EDICION' then '"OK_EDICION"' fi[/pft]>

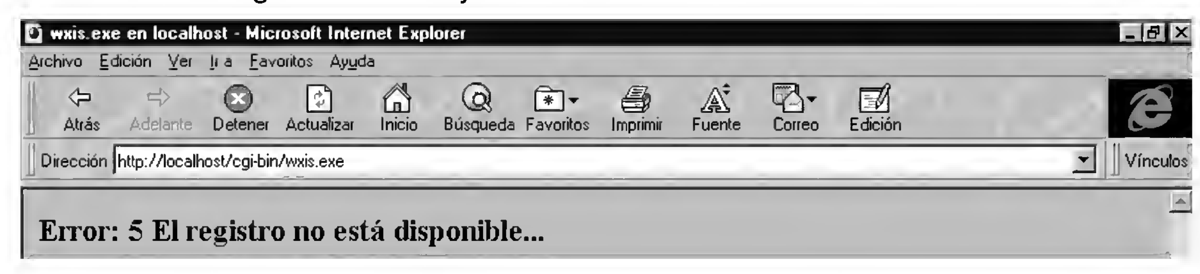
<p align="center"><font size="5"><em><strong><u>
   [pft]if v5001 = 'NUEVO' then 'Nuevo Registro' fi[/pft]
   [pft]if v5001 = 'BORRAR' then 'Eliminaci&oacute;n del Registro' fi[/pft]
   [pft]if v5001 = 'EDICION' then 'Edici&oacute;n de Registro' fi[/pft]
</u></strong></em></font></p>
```

Ahora **opción** asume el valor OK_EDICION. Con este nuevo valor pasará a la **script6.xis**.

La pantalla que se visualiza es:



Si en ese mismo momento otro usuario intentara editar el mismo registro, se visualiza el siguiente mensaje en el browser:



Luego de modificar los datos del registro y al hacer click sobre el botón OK, se envían los datos a la **script6.xis** pero esta vez con el valor OK_EDICION en la variable **opcion**. La script se ejecutarán las siguientes líneas:

```
    <!--Aqui comienza la opcion OK_EDICION -->
    <!-- En este punto, el flujo de información ha vuelto del formulario donde se
    tuvo la posibilidad de modificar el contenido del registro. Entonces, lo que
    resta en las líneas siguientes es actualizar físicamente el registro (grabarlo)
    -->
50. <label>OK_EDICION</label>
51. <do task=update>

    <!-- aqui se debe decir lo mismo que se colocó  en el label EDICION,
    después del task y antes del update ->
    <!--se va a realizar una tarea del tipo actualización, verificando si el registro
    no esta bloqueado. Se debe que indicar la base, el mfn y la llave -->
52.     <parm name=db>cds</parm>
53.     <parm name=mfn><pft>v6001</pft></parm>
54. <parm name=lockid><pft>v5002</pft></parm>

55.             <!-- Antes de grabar, compara esta llave con la que posee el registro
    en el campo v1101, si coinciden podrá actualizar el registro, sino se
```

mostrará un mensaje de error. Esta comparación corre por cuenta del wxis al momento de la grabación, liberando al programador de agregar líneas de programa para considerar esta situación

```
    <!-- se define el tag que se utilizará para verificar si la operación pudo
    realizarse -->
56.  <field action=define tag=1102>Isis_Status</field>
57.  <field action=define tag=1101>Isis_Lock</field>
58.  <update>

        <!-- antes de actualizar los datos, es necesario borrar los campos del
        registro que se han modificado; pero no el registro. Eso lo realizan las
        siguientes lineas. En este caso se ha procedido a borrar cada etiqueta,
        pero existe una modalidad para borrar todos los campos del registro -->
59.     <field action=delete tag=24>ALL</field>
60.     <field action=delete tag=70>ALL</field>
61.     <field action=delete tag=50>ALL</field>

        <!--se toma el titulo modificado -->
62.     <field action=cgi tag=24>titulo</field>
        <!--se toma el/los autores modificados -->
63.     <field action=cgi tag=70>autor</field>

        <!-- Como el campo autor es repetible, entonces se realiza el siguiente
        tratamiento: se reemplaza el campo 70 por un formato en el que se
        indica un nuevo argumento en el field con el parámetro split, que le
        indica que cada vez que encuentre un CR, lo reemplace por una nueva
        ocurrencia. -->
64.     <field action=replace tag=70 split=occ><pft>(v70/)</pft></field>

        <!--se toma el campo notas modificado-->
65.     <field action=cgi tag=50>notas</field>

        <!--ahora se procede a grabar el registro físicamente indicándose que
        luego, se lo desbloquee. Si se requiere dejar bloqueado, se debe definir
        el tiempo de expiración  -->
66.     <write>Unlock</write>

        <!-- a continuación se evalua la operación. La siguiente linea se debe
        incluir dentro del scope </update> debido a que en resultados.html se
        evalúa la variable 1102 que fuera del scope no se podría evaluar (no se
        ve) -->
67.     <display><htmlpft><pft>cat('ar/resultados.html')</pft></htmlpft></display
        >

68.  </update>
69.</do>
70.<flow action=exit>1</flow>
    <!--Aqui finaliza la opción OK_EDICION -->
```

En el código anterior se vio el borrado de cada una de las etiquetas de la siguiente forma:

```
<field action=delete tag=24>ALL</field>
<field action=delete tag=70>ALL</field>
<field action=delete tag=50>ALL</field>
```

Se podría realizar el borrado de todas ellas con una sola marca:

```
<field action=delete tag=list>ALL</field>
```

donde **list** hace referencia a todos los campos.??????

NOTAS IMPORTANTES:

- Después de verificar si el registro está bloqueado, se podrían utilizar varios display para tratar los estados del registro o bien utilizar las marcas *<Select case...>*.
- Dentro del Scope se tiene acceso a la información a nivel del Registro. Si se necesita algún dato que no es del registro, debe ser importado dentro del Scope.

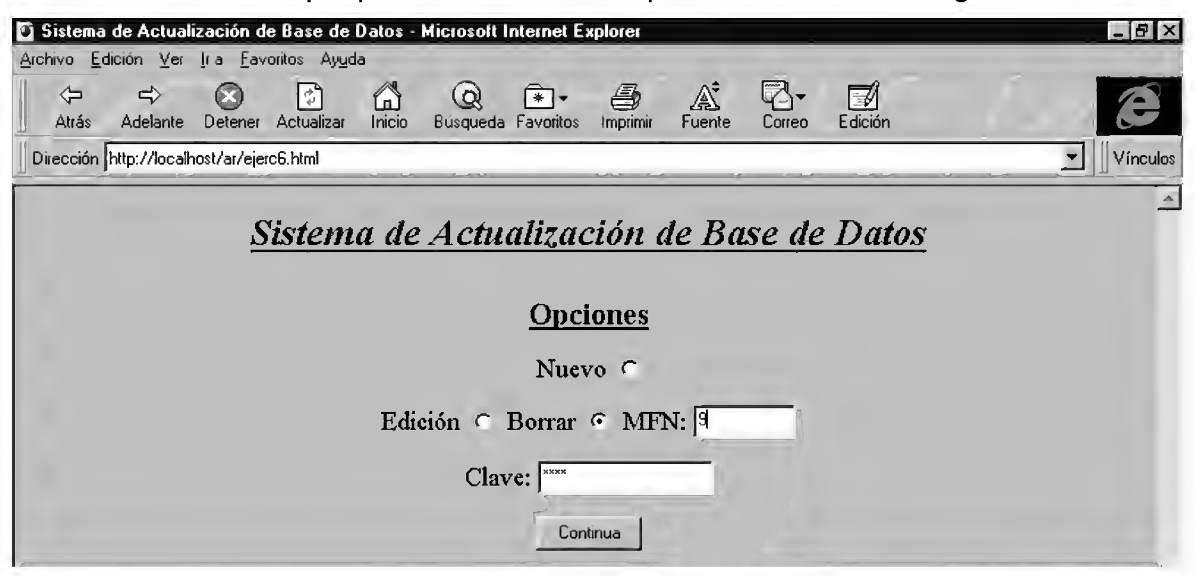
  Las variables de actualización automática que son definidas con *<field action=define* .....> pueden ser invocadas únicamente dentro del Scope, ya que son parte de los campos del registro. Después del mismo siguen estando disponibles las variables definidas fuera del Scope-registro.

  Siempre que se actualiza tener en cuenta: cuando se toma el registro para su modificación es **necesario** indicar el tiempo que tiene disponible el usuario para actualizar el registro. Si no se indica, el mismo queda por tiempo indefinido bloqueado y a disposición del usuario.
- Tener en cuenta que los campos importados al Scope son tomados como parte del registro y al momento de grabar son incorporados al mismo como campos nuevos. En el caso que se edite el mismo registro se generarán nuevas ocurrencias de estos campos. Lo conveniente es antes del proceso de grabado, realizar un **borrado** de todos ellos (CGI y los Editados en el formulario).

Borrado de Registros (BORRAR)

Para el borrado de registros, el procedimiento es similar al visto en EDICION. La diferencia sustancial radica en la acción de la marca *<write>*, que para este caso el parámetro es **Delete**.

El formulario que presenta el browser para el borrado del registro 9 es:



Cuando se hace click en el botón **Continúa** se llama a la **script6.xis** pasando el valor BORRAR en la variable ***opcion.***

Cuando la Script recibe el control, el flujo pasa al rótulo BORRAR cuya codificación es la siguiente:

```
        <!--Aqui comienza la opcion BORRAR -->
71.     <label>BORRAR</label>

        <!--la tarea que se realiza es de actualización, por ello update -->
72.     <do task=update>

              <!--se indica la base, el número de mfn (tag 6001) y la llave (tag 5002)
              -->
73.           <parm name=db>cds</parm>
74.           <parm name=mfn><pft>v6001</pft></parm>
75.           <parm name=lockid><pft>v5002</pft></parm>

              <!-- se indica el tiempo total del que dispone el cliente  para borrar el
              registro (60 segundos) -->
76.           <parm name=expire>60</parm>

              <!-- se define el  tag para verificar el estado de registro -->
77.           <field action=define tag=1102>Isis_Status</field>
```

```
<!-- se define un nuevo tag 1101 para dejar grabado en el registro la
información del bloqueo. Este tipo de campo no precisa ser importado
al scope-registro -->
<field action=define tag=1101>Isis_Lock</field>

<update>
   <!-- es necesario importar la variable opcion, que esta en el tag
   5001 para que se vea; y luego poder pasarla al formulario -->
   <field action=import tag=5001>5001</field>

   <!-- también es necesario importar la clave, para pasarla luego al
   formulario -->
   <field action=import tag=5002>5002</field>

   <!-- lo mismo para el MFN para luego pasarlo al formulario -->
   <field action=import tag=6001>6001</field>

   <!-- en la próxima instrucción se intenta únicamente bloquear el
   registro con la llave que se posee. De lograrse, el registro estará
   disponible por 60 segundos para ser eliminado por ese cliente. -->
   <write>Lock</write>

   <!-- se indaga si se ha bloqueado el registro a través del Isis_Status
   que corresponde a la etiqueta 1102 (val(v1102) <> 0); y de estar
   bloqueado por otro usuario, se emite una leyenda de error -->
   <display><pft>If val(v1102) <> 0 then '<h3><b>Error: ',v1102,'   El
   registro no est&aacute; disponible...</b></h3>' fi</pft></display>

   <!-- como el registro no esta bloqueado (val(v1102) = 0), se llama al
   formulario que presentará la información del registro con la opción
   de eliminarlo-->
   <display><htmlpft><pft>if val(v1102) = 0 then
   cat('ar/formulario.html') fi</pft></htmlpft></display>

</update>
</do>
<flow action=exit>1</flow>
<!-- aqui finaliza la opción BORRAR -->
```

A **formulario.html** se lo llama pasándoles el nombre de la base de datos, el mfn que se actualiza, la llave del usuario (estos datos son necesarios pasarlos porque luego los necesitará la **script6.xis** para proceder al borrado del registro) y el valor BORRAR en **opcion.**

Recordar que el formulario indaga el valor de **opcion** y dependiendo de él, realiza la actualización de la variable y también el título que se visualiza varía:

```
<input type="hidden" name="opcion" value=
   [pft]if v5001 = 'NUEVO' then '"OK_NUEVO"' fi[/pft]
   [pft]if v5001 = 'BORRAR' then '"OK_BORRAR"' fi[/pft]
   [pft]if v5001 = 'EDICION' then '"OK_EDICION"' fi[/pft]>
```

```
<p align="center"><font size="5"><em><strong><u>
   [pft]if v5001 = 'NUEVO' then 'Nuevo Registro' fi[/pft]
   [pft]if v5001 = 'BORRAR' then 'Eliminaci&oacute;n del Registro' fi[/pft]
   [pft]if v5001 = 'EDICION' then 'Edici&oacute;n de Registro' fi[/pft]
</u></strong></em></font></p>
```

Ahora **opcion** asume el valor OK_BORRAR. Con este nuevo valor pasará a la **script6.xis**.

La pantalla que se visualiza es:

Nuevo Registro - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

Atrás Adelante Detener Actualizar Inicio Búsqueda Favoritos Imprimir Fuente Correo Edición

Dirección http://localhost/cgi-bin/wxis.exe Vínculos

Eliminación del Registro

Titulo: A New type of porometer for measuring stomatal openi

Autor:

Notas: Incl. bibl.

OK

Si en ese mismo momento otro usuario intentara editar el mismo registro, se visualiza el siguiente mensaje en el browser:



Notar que el ISIS_STATUS contiene el valor 5 cuando el registro se encuentra bloqueado.

Luego al hacer click sobre el botón OK, se envían los datos a la **script6.xis** pero esta vez con el valor OK_BORRAR en la variable **opción**.

```
<!--Aqui comienza la opción OK_BORRAR -->
   <!-- En este punto, el flujo de información ha vuelto del formulario donde se
   tuvo la posibilidad de visualizar el contenido del registro antes de borrarlo.
   Entonces, lo que resta en las líneas siguientes es borrar lógicamente el
   registro -->
<label>OK_BORRAR</label>
<do task=update>
```

```
<!--se va a realizar una tarea del tipo actualización, verificando si el registro
no esta bloqueado. Se debe indicar la base, el mfn y la llave -->
<parm name=db>cds</parm>
<parm name=mfn><pft>v6001</pft></parm>
<parm name=lockid><pft>v5002</pft></parm>

<!-- se define el tag que se utilizará para verificar si el registro sigue
disponible para este usuario, porque puede suceder que el tiempo haya
expirado.-->
<field action=define tag=1102>Isis_Status</field>
<field action=define tag=1101>Isis_Lock</field>
<update>

    <!-- antes de borrar el registro, es necesario borrar todos los campos -->
    <field action=delete tag=list>ALL</field>

    <!--ahora se procede a borrar  el registro -->
    <write>Delete</write>

    <!-- a continuación se evalúa la operación. La siguiente línea se debe
    incluir dentro del scope </update> debido a que en resultados.html se
    evalúa la variable 1102 que fuera del scope no se podría evaluar (no se
    ve) -->
    <display><htmlpft><pft>cat('ar/resultados.html')</pft></htmlpft></display
    >
</update>
</do>
<flow action=exit>1</flow>
<!--Aqui finaliza la opción OK_BORRAR -->

</section>
</IsisScript>
```

A continuación se transcribe el código completo de la **script6.xis:**

```
<IsisScript>
  <section>
    <display><pft>'Content-type: text/html'/#</pft></display>
    <field action=cgi tag=5000>IsisScript</field>
    <field action=cgi tag=5001>opcion</field>
    <field action=cgi tag=6001>id</field>
    <field action=add tag=3030><pft>'c:\curso\'</pft></field>
    <parm name=cipar>
    <pft>'cds.*='v3030'cds.*'/</pft></parm>
    <flow action=jump><pft>v5001</pft></flow>

    <!--Aqui comienza la opción NUEVO -->
```

```
<label>NUEVO</label>
<display><htmlpft><pft>cat('ar/formulario.html')</pft></htmlpft></display>
<flow action=exit>1</flow>
<!-- aqui finaliza la opción NUEVO -->

<!--Aqui comienza la opción OK_NUEVO  en la script6.xis-->
<label>OK_NUEVO</label>
<do task=update>
     <parm name=db>cds</parm>
     <parm name=mfn>New</parm>
     <parm name=fst><pft>cat('cds.fst')</pft></parm>
     <field action=define tag=1102>Isis_Status</field>
     <field action=define tag=1101>Isis_Lock</field>
     <update>
        <field action=cgi tag=24>titulo</field>
        <field action=cgi tag=70>autor</field>
        <field action=replace tag=70 split=occ><pft>(v70/)</pft></field>
        <field action=cgi tag=50>notas</field>
        <write>Unlock</write>
        <display><htmlpft><pft>cat('ar/resultados.html')</pft></htmlpft></displ
ay>
     </update>
</do>
<flow action=exit>1</flow>
<!-- aqui finaliza la opción OK_NUEVO -->

<!--Aqui comienza la opción EDICION -->
<label>EDICION</label>
<do task=update>
   <parm name=db>cds</parm>
   <parm name=mfn><pft>v6001</pft></parm>
   <parm name=lockid><pft>v5002</pft></parm>
   <parm name=expire>60</parm>
   <field action=define tag=1102>Isis_Status</field>
   <field action=define tag=1101>Isis_Lock</field>
   <update>
        <field action=import tag=5001>5001</field>
        <field action=import tag=5002>5002</field>
        <field action=import tag=6001>6001</field>
        <write>Lock</write>
        <display><pft>If val(v1102) <> 0 then '<h3><b>Error: ',v1102,'   El
registro
         no est&aacute; disponible...</b></h3>' fi</pft></display>
        <display><htmlpft><pft>if val(v1102) = 0 then cat('ar/formulario.html')
        fi</pft></htmlpft></display>
   </update>
</do>
<flow action=exit>1</flow>
<!-- aqui finaliza la opcion EDICION -->
```

```
<!--Aqui comienza la opcion OK_EDICION -->
<label>OK_EDICION</label>
<do task=update>
        <parm name=db>cds</parm>
        <parm name=mfn><pft>v6001</pft></parm>
        <parm name=lockid><pft>v5002</pft></parm>
        <field action=define tag=1102>Isis_Status</field>
        <field action=define tag=1101>Isis_Lock</field>
        <update>
              <field action=delete tag=24>ALL</field>
              <field action=delete tag=70>ALL</field>
              <field action=delete tag=50>ALL</field>
              <field action=cgi tag=24>titulo</field>
              <field action=cgi tag=70>autor</field>
              <field action=cgi tag=50>notas</field>
              <field action=replace tag=70 split=occ><pft>(v70/)</pft></field>
              <write>Unlock</write>
              <display><htmlpft><pft>cat('ar/resultados.html')</pft></htmlpft>
              </display>
        </update>
</do>
<flow action=exit>1</flow>
<!--Aqui finaliza la opción OK_EDICION -->

<!--Aqui comienza la opción BORRAR -->
<label>BORRAR</label>
<do task=update>
      <parm name=db>cds</parm>
      <parm name=mfn><pft>v6001</pft></parm>
      <parm name=lockid><pft>v5002</pft></parm>
      <parm name=expire>60</parm>
      <field action=define tag=1102>Isis_Status</field>
      <field action=define tag=1101>Isis_Lock</field>
      <update>
            <field action=import tag=5001>5001</field>
            <field action=import tag=5002>5002</field>
            <field action=import tag=6001>6001</field>
            <write>Lock</write>
            <display><pft>If val(v1102) <> 0 then '<h3><b>Error: ',v1102,'   El
            registro no est&aacute; disponible...</b></h3>' fi</pft></display>
            <display><htmlpft><pft>if val(v1102) = 0 then cat('ar/formulario.html')
            fi</pft></htmlpft></display>
      </update>
</do>
<flow action=exit>1</flow>
<!-- aqui finaliza la opción BORRAR -->

<!--Aqui comienza la opción OK_BORRAR -->
<label>OK_BORRAR</label>
<do task=update>
```

```
            <parm name=db>cds</parm>
            <parm name=mfn><pft>v6001</pft></parm>
            <parm name=lockid><pft>v5002</pft></parm>
            <field action=define tag=1102>Isis_Status</field>
            <field action=define tag=1101>Isis_Lock</field>
            <update>
                <field action=delete tag=list>ALL</field>
                <write>Delete</write>
                <display><htmlpft><pft>cat('ar/resultados.html')</pft></htmlpft>
                </display>
            </update>
        </do>
        <flow action=exit>1</flow>
        <!--Aqui finaliza la opción OK_BORRAR -->

    </section>
</IsisScript>
```

Funciones

A los efectos de conocer la aplicación de las funciones, se crea una nueva Script denominada **script7.xis**.

Las funciones deben ser declaradas entre las marcas *<IsisScript>* y *<section>*; es decir al comienzo de la Script.

```
<IsisScript>
        Aquí va la declaración de la función.
        <section>
        .....
```

Se ha visto que el **loop** crea un Scope; la función también crea un Scope diferente a la de la *section*.

El formato o sintaxis de la función es:

```
<function name=UNO>  (name es uno de los parámetros)
        ...... líneas de ejecución
</function>
```

Para llamar a la función en la Section, se debe utilizar la marca *<call name= ....>*

La sintaxis es:

```
<call name=Uno>Parámetro_que_se_pasa_a_la_función</call>
```

Lo que se coloca entre las marcas es el parámetro que se le pasa a la función.

Se puede definir una función con pasaje de parámetros. Para ello se define:

```
<function name=DOS action=replace tag=100>
```

donde los parámetros que se reciben son guardados en el tag 100, que es una variable del scope-function.

Se ha confeccionado un formulario que llama la **script7.xis** donde se observa la aplicación de las funciones. El código del formulario es el siguiente:

```
<html>
    <head>
        <title>Funciones</title>
    </head>
    <body>
    <H3>Aplicación de Funciones</h3>
    <form action="/cgi-bin/wxis.exe" method="post">
        <input type="hidden" name="IsisScript" value="ar/script7.xis">
        <p><input type="submit" value="Continua"> </p>
        </form>
    </body>
</html>
```

La visualización en pantalla es la siguiente:



Al hacer click en **Continua**, se llama a la **script7.xis** cuyo código es el siguiente:

```
<IsisScript>
    <!-- La función UNO no recibe parámetros, imprime un mensaje y retorna -->
    <function name=UNO>
        <display><pft>'<h2>Este es el mensaje que imprime la FUNCION
        UNO</h2><br>'</pft></display>
    </function>
```

```
<!-- La función DOS recibe 1 parámetro (valor Wxis), que lo coloca en el tag
100, imprime un mensaje junto al valor de la etiqueta y retorna -->
<function name=DOS action=replace tag=100>
      <display><pft>'<hr><h3>Funci&oacute;n DOS valor pasado:
      ',v100,'</h3>'</pft></display>
</function>

<!-- La función TRES recibe varios parámetros (cada uno de ellos se coloca
en una ocurrencia distinta dentro del tag 100. Se crean tantas ocurrencias
como parámetros se pasen) ello se logra con el atributo split. La función
imprime un mensaje junto al valor de la etiqueta y retorna. Notar que el
mensaje está entre paréntesis para hacer el tratamiento de cada una de las
ocurrencias -->
<function name=TRES action=replace tag=100 split=occ>
      <display><pft>('<h3>Funci&oacute;n TRES, valor pasado: ',v100,'
      <i>Ocurrencia: ',f(iocc,3,0),'</i></h3>)</pft></display>
</function>
<!-- La función CUATRO recibe un parámetro que lo coloca en la etiqueta
100, ese valor es reemplazado por un cálculo (es por ello que se utiliza field
action=replace). Imprime un mensaje junto al valor de la etiqueta, antes de
retornar coloca el valor del calculo que esta en la etiqueta 100, nuevamente
en la etiqueta 2000. Esto se logra mediante una exportación. Este es el
caso en que la función retorna un valor  -->
<function name=CUATRO action=replace tag=100>
      <display><pft>'<h3>Funci&oacute;n CUATRO, valor pasado:
      ',v100,'</h3>'</pft></display>
      <field action=replace tag=100><pft>f(val(v100) * 10,3,0)</pft></field>
      <display><pft>'<h3>Funci&oacute;n CUATRO, valor calculado:
      ',v100,'</h3>'</pft></display>
      <return action=export tag=2000>100</return>
</function>

<section>
   <display><pft>'Content-type: text/html'/#</pft></display>
   <display><pft>’<h1><u>P&aacute;gina sobre
   funciones</u></h1><hr>'</pft></display>

   <!-- se llama a la función UNO, lo que está entre las marcas call, es el
   parámetro que  se pasa. En este caso, el parámetro se pasa pero no
   tiene incidencia en la función. -->
   <call name=UNO>parametro_que_no_se_utiliza</call>
   <display><pft>'<h4>Mensaje de Retorno: Funci&oacute;n
   UNO</h4><p>'</pft></display>

   <!-- se llama a la función DOS, se le pasa el parámetro Wxis. -->
   <call name=DOS><pft>'Wxis'</pft></call>
   <display><pft><h4>Mensaje de Retorno: Funci&oacute;n
   DOS</h4><hr><p>'</pft></display>
```

```
    <!-- se llama a la función TRES, se le pasa varios parámetros, cada uno
    de ellos separados por / y dentro de las marcas <pft> -->
    <call name=TRES><pft>'Wxis1'/'Wxis2'/'Wxis3'</pft></call>
    <display><pft><h4>Mensaje de Retorno: Funci&oacute;n
    TRES</h4><hr>'</pft></display>

    <!-- se llama a la función CUATRO, se le pasa el valor 30 que
    previamente se asigna a la etiqueta 2000 -->
    <field action=add tag=2000>30</field>
    <display><pft>'<h3>Valor de v2000 antes de la funci&oacute;n:
    ',v2000,'</h3>'</pft></display>
    <call name=CUATRO><pft>v2000</pft></call>
    <display><pft>'<h3>Valor de v2000 despu&eacute;s de la
    funci&oacute;n: ',v2000,'</h3>'</pft></display>
    <display><pft>'<h4>Mensaje de Retorno: Funci&oacute;n
    CUATRO</h4>'</pft></display>
    </section>
</IsisScript>
```

El browser visualiza la siguiente pantalla:

wxis.exe en localhost - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

Atrás Adelante Detener Actualizar Inicio Busqueda Favoritos Imprimir Fuente Correo Edición

Dirección http://localhost/cgi-bin/wxis.exe Vínculos

Página sobre funciones

Este es el mensaje que imprime la FUNCION UNO

Mensaje de Retorno: Funcion UNO

Función DOS valor pasado: Wxis

Mensaje de Retorno: Funcion DOS

Función TRES, valor pasado: Wxis1 *Ocurrencia: 1*

Función TRES, valor pasado: Wxis2 *Ocurrencia: 2*

Función TRES, valor pasado: Wxis3 *Ocurrencia: 3*

Mensaje de Retorno: Funcion TRES

wxis.exe en localhost - Microsoft Internet Explorer

Archivo Edición Ver Ir a Favoritos Ayuda

Atrás Adelante Detener Actualizar Inicio Búsqueda Favoritos Imprimir Fuente Correo Edición

Dirección http://localhost/cgi-bin/wxis.exe Vínculos

Valor de v2000 antes de la función: 30

Función CUATRO, valor pasado: 30

Función CUATRO, valor calculado: 300

Valor de v2000 después de la función: 300

Mensaje de Retorno: Funcion CUATRO

